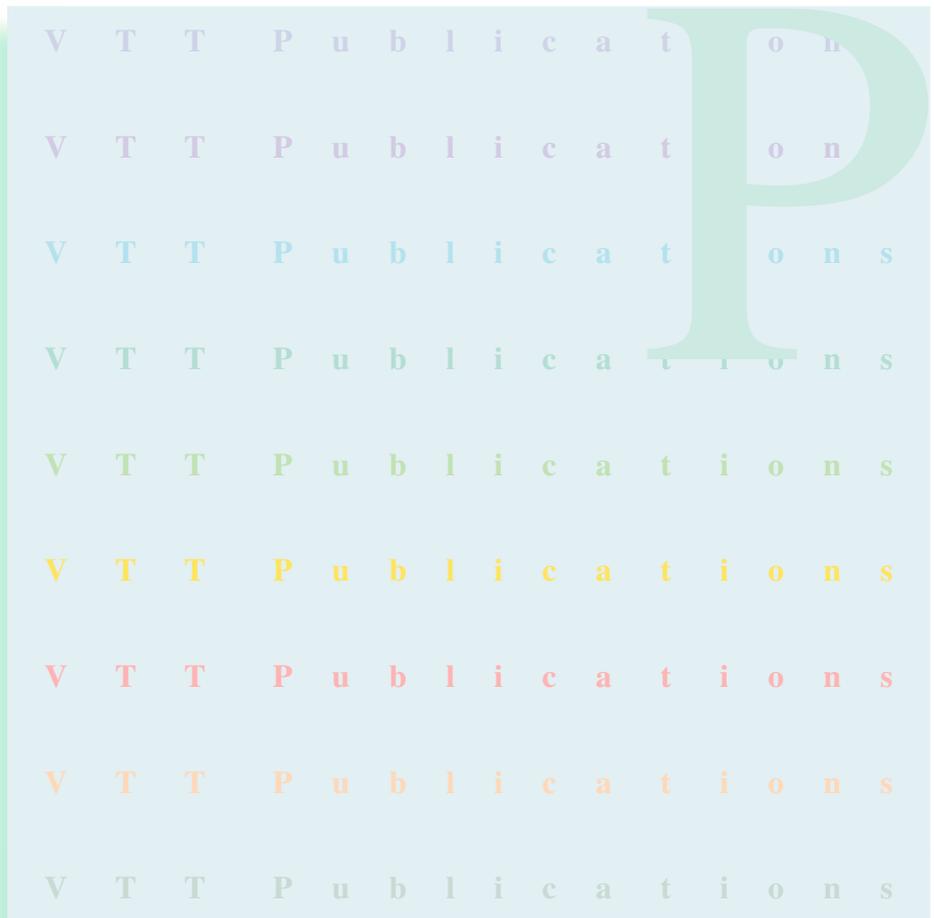


Tua Rahikkala

Towards virtual software configuration management

A case study



VTT PUBLICATIONS 409

Towards virtual software configuration management

A case study

Tua Rahikkala

VTT Electronics

*Academic Dissertation to be presented with the assent of the Faculty of Science,
University of Oulu, for public discussion in Auditorium GO101, on May 30th, 2000, at
12 noon.*



TECHNICAL RESEARCH CENTRE OF FINLAND
ESPOO 2000

ISBN 951-38-5567-8 (soft back ed.)
ISSN 1235-0621 (soft back ed.)

ISBN 951-38-5568-6 (URL: <http://www.inf.vtt.fi/pdf/>)
ISSN 1455-0849 (URL: <http://www.inf.vtt.fi/pdf/>)

Copyright © Valtion teknillinen tutkimuskeskus (VTT) 2000

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

Technical Research Centre of Finland (VTT), Vuorimiehentie 5, P.O.Box 2000, FIN-02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektroniikka, Sulautetut ohjelmistot, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Inbyggd programvara, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN-90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Technical editing Leena Uksskoski

Otamedia Oy, Espoo 2000

Rahikkala, Tua. Towards virtual software configuration management. A case study. Espoo 2000, Technical Research Centre of Finland, VTT Publications 409. 110 p. + app. 57 p.

Keywords Software Configuration Management (SCM), Virtual Software Corporation (VSC), distributed teamwork environment, dynamic teamwork environment, Software Process Improvement (SPI), SCM process analysis

Abstract

The organisational performance of software companies has become critically important. While customer requirements are changing and varying ever more frequently, and an effective management of the software process is becoming more and more essential, the appropriateness of current software development models has become questionable. One of the current global trends in software development is transorganisational collaborative work in distributed, dynamic teamwork environments, called *Virtual Software Corporations (VSC)*. The virtual environment presents particular challenges to *Software Configuration Management (SCM)*, while the rate of change concerning the size, complexity and duration of software projects is increasing constantly. While there is a clear need for VSCs to analyse and improve their SCM processes, they may approach the subject from rather different angles. Generally, when analysing the current status of the software process, it is vital to understand what the context of the process is and what activities it includes. On the other hand, our study makes it evident that VSCs present new kinds of challenges to SCM that can not be solved by means of traditional SCM procedures and techniques only. Furthermore, many of the VSC variants are likely to realise that the increasing complexity and number of their software processes also affect the SCM process directly. Hence, a sound understanding of the specific context is required for analysing and improving the SCM in a VSC. Gaining extensive knowledge of the SCM-related requirements for industrial VSCs can thus be considered a prerequisite for enhancing the process description. This dissertation presents an approach for defining the SCM requirements for a VSC. On the basis of a requirements analysis, an expanded SCM process description is introduced. As a result, this dissertation introduces an enhanced approach to SCM process analysis and improvement for the distinct context of a VSC environment.

Preface

The main research of this thesis was carried out between October 1997 and February 2000 during the EU/Espirit project #25754 VISCOUNT at VTT Electronics. The VISCOUNT project was funded by the European Commission, the National Technology Agency of Finland (Tekes), and VTT Electronics. Here, I would like to extend my gratitude to the VISCOUNT project team for valuable and fruitful co-operation, especially to Mr. Antti Välimäki for his collaboration during the study and to Dr. Jorma Taramaa for the significant encouragement and support I have received from him. I also wish to thank the Jenny and Antti Wihuri Foundation and Seppo Säynäjäkangas Foundation. Their financial support enabled me to concentrate on my thesis during the most intensive writing period.

I wish to dedicate my appreciation to Prof. Veikko Seppänen who has tirelessly reviewed several drafts of my texts and who has kept spurring me on, as well as to Prof. Jouni Similä, the supervisor of this thesis, for his support and helpful comments. Furthermore, I would like to express my thanks to Dr. Cornelia Boldyreff (University of Durham, UK) and Dr. Richard Foley (Glasgow Caledonian University, Scotland) for providing their valuable comments and recommendations as the nominated reviewers of the thesis. I also wish to thank Mr. Seppo Keränen, who has proofread most of my texts. More broadly, my thanks for their stimulating influence over the years go to my friends and colleagues in my previous places of employment and at VTT Electronics.

Finally, I wish to express my gratitude to my family for all the support I have received, especially to my father, Matti Rahikkala, who has always been motivating and encouraging me in my studies.

Glasgow, April 2000

Tua Rahikkala

List of Original Papers

This thesis includes six original papers published in the proceedings of international conferences. The papers are included in the thesis with the permission of their original publishers.

- I Rahikkala, T., Taramaa, J., Välimäki, A. 1998. Industrial Experiences from SCM Current State Analysis. In: Magnusson, B. (ed.). System Configuration, System Configuration Management, ECOOP 98 SCM-8. Brussels, Belgium: Springer-Verlag. Pp. 13–25. (LNCS 1439.) ISBN 3–540–64733–3.
- II Rahikkala, T., Taramaa, J., Välimäki, A. 1998. Experiences from Goal Definition for SCM Process Improvement. In: Hawkins, C., Ross, M., Staples, G., Thompson, J.B. (eds.). INSPIRE III Process Improvement Through Training and Education. London, UK: The British Computer Society. Pp. 58–68. ISBN 1–902–50503–4.
- III Rahikkala, T., Nättinen, M., Välimäki, A. 1998. Practical Experiences from SCM Measurement within a Distributed Environment. In: 10th Anniversary Software Measurement in Practice, The United Kingdom Software Metrics Association. London, UK: UKSMA'98. 10 p.
- IV Rahikkala, T., Blackwood, R., Cocchio, L., Gray, E., Kucza, T., Newman, J., Välimäki A. 1998. Experiences from Requirements Analysis for SCM Process Improvement in Virtual Software Corporations. In: Proceedings of the European Conference on Software Process Improvement. SPI'98. 7 p.
- V Nättinen, M., Rahikkala, T., Välimäki, A. 1999. Experiences from the Pilot Operation and Commissioning Phase of an SCM Process Improvement Program. In: Proceedings of the 25th EUROMICRO Conference, EUROMICRO 99, Volume II. Milan, Italy: IEEE Computer Society. Pp 185–191. ISBN 0–795–0321–7.

- VI Ronkainen, J., Blackwood, R., Rahikkala, T. 1999. Automating SCM Metric Data Collection and Analysis in Virtual Software Corporations. In: Proceedings of the 25th EUROMICRO Conference, EUROMICRO 99, Volume II. Milan, Italy: IEEE Computer Society. Pp 279–283. ISBN 0–795–0321–7.

The papers will be referred to in the text by the corresponding Roman numerals (I–VI).

The author of this dissertation is the principal author of the papers I–IV. The research ideas of these papers have been presented by the author. The author has carried out most of the practical work presented in the papers. For papers I–II Dr. Jorma Taramaa has been working as an advisor, providing his expertise in process improvement projects. For papers I–V Mr. Antti Välimäki has given his comments based on his view on industrial organisations. In paper IV, the other authors have been asked to give their comments from the viewpoint of the organisations they represented.

In paper V, the principal author is Ms. Minna Nättinen and in paper VI Mr. Jussi Ronkainen. However, the author of this dissertation provided new research ideas and her expertise for the papers, and the research work presented in the papers has been managed by her. Furthermore, the author of this dissertation has significantly contributed to the papers in terms of planning and developing case studies, as well as co-writing the papers.

Contents

Abstract.....	3
Preface	4
List of Original Papers.....	5
List of Names and Acronyms	10
1. Introduction.....	11
1.1 Background.....	11
1.2 Research Problem.....	14
1.3 Research Strategy	16
1.4 Outline of the Thesis and Main Results.....	18
2. Related Work	21
2.1 Introduction	21
2.2 Basic Concepts of Software Configuration Management (SCM)	22
2.3 SCM Process Definitions	23
2.4 Virtual Software Corporations and SCM	26
2.4.1 Introduction to Virtual Corporations.....	26
2.4.2 VSC Attributes and Characteristics.....	27
2.4.3 Summary of Virtual Software Corporations and SCM	32
2.5 Summary of Related Work	33
3. SCM Requirements Analysis in a VSC	35
3.1 Introduction	35
3.2 Case Environment	36
3.2.1 VISCOUNT project.....	36
3.2.2 Application Organisations.....	38
3.3 Defining the SCM Requirements in a VSC through the Case Study.....	41
3.3.1 Background	41
3.3.2 Use Cases - A Practical Method for SCM Requirements Analysis ..	42
3.3.3 Progress of SCM Requirements analysis	45
3.3.4 An Industrial Example of the SCM Requirements Analysis.....	47
3.4 Evaluating the Results	50

4. Towards Virtual SCM Process	54
4.1 Introduction	54
4.2 Lessons Related to the SCM Challenges	54
4.2.1 Background	54
4.2.2 SCM Improvement	57
4.2.3 SCM Teamwork	57
4.2.4 SCM Planning	62
4.3 The Expanded SCM Process Description	63
4.4 Conclusions	65
5. Developing SCM to meet VSC Requirements	66
5.1 Introduction	66
5.2 Improvement Framework	67
5.3 SCM Current State Analysis	69
5.3.1 General	69
5.3.2 Applying an Expanded SCM Description in SCM-Pr ² imer	70
5.4 The Goals for Improved SCM	75
5.5 Plan for SCM Improvement Steps	77
5.6 Pilot Operation and Commissioning	78
5.7 Conclusions	79
6. Validation	81
6.1 Introduction	81
6.2 Validation of the Approach and Methods	83
6.2.1 Internal Validity	83
6.2.2 External Validity	83
6.2.3 Reliability	84
6.3 Construct Validity	86
6.3.1 Changes to be Studied	87
6.3.2 Validation Process	90
6.3.3 Evaluation of the SCM Improvement Program at Neles	91
6.3.4 Conclusions	94
7. Conclusions	96
7.1 Research Results	96
7.2 Future Research	98

8. Introduction to the Papers	100
8.1 Paper I.....	101
8.2 Paper II	101
8.3 Paper III.....	102
8.4 Paper IV.....	102
8.5 Paper V	103
8.6 Paper VI.....	103

References.....	104
-----------------	-----

Appendices

Papers I–VI

***Appendices of this publication are not included in the PDF version.
Please order the printed version to get the complete publication
(<http://otatrip.hut.fi/vtt/jure/index.html>)***

List of Names and Acronyms

BA	British Aerospace (http://www.bae.co.uk)
Bootstrap	European software process assessment and improvement methodology (http://www.bootstrap-institute.com/)
ClearCase®	SCM tool, developed by Rational Software Corporation Inc. (http://www.rational.com)
CM	Configuration Management
CMM	Capability Maturity Model (http://www.sei.cmu.edu/cmm)
debis	debis Systemhaus GEI (http://www.debis.com)
GCU	Glasgow Caledonian University (http://www.gcal.ac.uk)
GQM	Goal-Question-Metric method (e.g. http://www.gqm.nl)
IEEE	Institute of Electrical and Electronics Engineers (http://www.ieee.org)
ISO	The International Standardization Organization (http://www.iso.ch)
LIFESPAN™	SCM tool, developed by BA Ltd. (http://www.lifespan.co.uk)
Pr ² imer	Practical Process Improvement for Embedded Real-Time Software, developed by VTT Electronics
SCI	Software Configuration Item
SCM	Software Configuration Management
SCM-Pr ² imer	SCM focused software process improvement approach of Pr ² imer.
SEI	The Software Engineering Institute
SIA	Società Italiana Avionica (http://www.sia-av.it)
SPI	Software Process Improvement
SPICE	Software Process Improvement and Capability dEtermination (e.g. http://www.iese.fhg.de/SPICE)
Tekes	National Technology Agency of Finland (http://www.tekes.fi)
Neles	Neles Automation Ltd. (http://www.nelesautomation.com)
VISCOUNT	Virtual Software Corporation Universal Testbed, a project of EU/Esprit programme #25754 (http://www.viscount.org)
VSC	Virtual Software Corporation
VTT	Technical Research Centre of Finland (http://www.vtt.fi)

1. Introduction

1.1 Background

The importance of software engineering as a part of product development has been increasing continuously in the last few years. One of the most important reasons for this can be found in the emphasis laid on quality improvement and on reducing the delivery time to the market. At the same time, the rate of change concerning the size, amount, and complexity of software incorporated in products has become faster and faster, and the development costs have continued to rise (Brown 1996, Fitzgerald & O'Kane 1999). The various changes in, and customer-specific versions of, products are typically brought about by the development of new software features. For instance, software has become one of the core technologies in various embedded systems applications in which new product features can be more easily and efficiently implemented by means of software (Seppänen et al. 1996). Furthermore, software implementation is sometimes the only possibility for reasons of cost-efficiency and project schedule.

Today, it has become difficult for a single organisation to provide all the necessary expertise required for developing complex systems. Instead of this, major development projects frequently have to be carried out through co-operation of several organisations, each bringing their own areas of expertise to the project (Boldyreff et al. 1996, Noll & Scacchi 1997). On the other hand, several organisations have extended their own activities into various new areas due to the lack of resources, or for social, commercial, or economic reasons (Alstyn van 1997). As a result of the various changes, the traditional single site software development has become unsuitable for many organisations, and the need for distributing professional and specialised resources has increased (Venkatraman & Henderson 1998, Asklund et al. 1999). Therefore, organisations are getting increasingly involved in software development efforts that are networked and distributed in nature. Under such circumstances, the importance of software process management in distributed teamwork is becoming increasingly evident.

As stated above, the organisational performance of software companies has become critically important. While customer requirements are changing and

varying ever more frequently, and an effective management of the software process is becoming more and more essential, the suitability of the current practices for software development has become questionable (Venkatraman & Henderson 1998, Van De Vanter & Murer 1999). One of the current global trends is transorganisational collaborative work in distributed, dynamic teamwork environments, called *Virtual Software Corporations (VSC)*. A VSC is a teamwork environment in which all members work together to achieve common interests or goals (Davidow & Malone 1992, Lipnack & Stamps 1997, Ahuja & Carley 1998). Typically, such teams are distributed in different locations and even across countries, and the bonds between them are temporary and sporadic - related to the specific projects or tasks. The teams may consist of employees, managers, suppliers, and customers. People may also be working simultaneously for several companies. While these companies might be collaborators on developing one product, they may also be competitors on another project, developed in a different network.

The primary task of *Software Configuration Management (SCM)* is to establish and to maintain product integrity in software projects throughout the life cycle of the project (Paulk et al. 1995). In the past, version models, system constructions, and system modelling have been the main issues in SCM, and in these areas a lot of research has been carried out (Conradi & Westfechtel 1996, Estublier & Casallas 1994). However, as a result of the changes in software development described above, the focus of the research has shifted towards other interests, one of the main trends being the support provided by SCM for VSCs (Noll & Scacchi 1997, Hoek van der et al. 1998). SCM has become a great challenge for VSCs, being one of the core processes in the management of collaboration within a distributed VSC environment. The challenge provided by SCM exhibits itself at several levels in VSCs.

At one level, the main issue is the support of concurrent software development tasks and the distribution of large amounts of data in a timely fashion over great distances (Buffenbarger & Gruell 1999). In these instances, the concern of the organisation may focus on technical areas such as distribution techniques, branching and merging strategy, interfaces to other SCM tools, data transmission security, and data compression and encryption (e.g. Noll & Scacchi 1997, Buffenbarger & Gruell 1999). Depending on the SCM techniques and tools used, these issues may prove to be considerable barriers to effective software

development. In general, many organisations are still trying to solve SCM problems mainly by acquiring tools that provide answers for the technical challenges they are facing. These organisations often have extensive requirements for such tools (Dart 1995). On one hand, the fact that the problems might not disappear by tool acquisition can be tracked back to the issue of not actually comprehending SCM as a process, which can be supported by the acquired tools (Wein et al. 1995, Frühauf & Zeller 1999). On the other hand, no single SCM system is usually capable of providing solutions to all problems and requirements an organisation may have (Asklund et al. 1999, Estublier 1999, Frühauf & Zeller 1999). Therefore, we would like to emphasise the importance of understanding SCM as a process, considering the fact that the process is becoming rapidly virtual, by being distributed over heterogeneous and temporary organisational structures and many different locations. Thus, when analysing and improving the SCM within a VSC, what is required is a shift from solving technical problems towards a more extensive process-based view.

Organisational matters may also present considerable challenges to SCM within a VSC. For example, there is the issue of managing human, technical, and process resources within an environment where the team members may come from greatly varying cultural backgrounds, and may be adhering to differing daily procedures and practices (Hoek van der et al. 1996, Noll & Scacchi 1997, Van De Vanter & Murer 1999). In many instances, making the communication and task co-ordination across different locations and time zones effective enough presents a tough practical organisational challenge to SCM. One of the main implications for software production organisations is that they need to carefully analyse the current status of the SCM process within the entire VSC, as well as to define and to maintain suitable and effective working methods for improving the SCM practices.

While VSCs decidedly need to analyse and improve their SCM processes, they may approach the subject from a number of different angles. A widely used framework for improvement actions is provided by present software process improvement (SPI) life cycle models. Typically, these models consist of at least three main activities: Evaluation, Planning and Effecting (Kinnula 1999). The evaluation phase, determining the current status of the process, establishes a baseline for the whole improvement cycle as well as for further work. Generally, when analysing the current status of the process it is vital to understand the

context of the process along with the activities included in it (Wein et al. 1995, Grady 1997, Taramaa 1998). However, our study demonstrates that VSCs do present new kinds of challenges to SCM unsolvable by means of traditional SCM procedures and techniques only. Furthermore, many of the VSC variants are likely to realise that the increasing complexity and number of their software business processes affect the SCM process directly. Hence, a sound understanding of the specific context is required for analysing and improving the SCM within a VSC. Gaining extensive knowledge of the SCM-related requirements for industrial VSCs can thus be considered a prerequisite for enhancing the process description. In this dissertation, an approach for defining SCM requirements within a VSC will be presented. On the basis of the requirements analysis, we are introducing an expanded SCM process description. Thus, the thesis presents an enhanced approach to SCM process analysis and improvement within the distinct context of a VSC environment.

1.2 Research Problem

The basic inspiration for the dissertation are the SCM-related problems appearing in industrial software development organisations operating within a virtual environment. The problems are critical for the industry and the solutions are, indisputably, complicated. The complexity arises from the varied nature of the VSCs. It is not possible to give a single specific definition to a VSC. VSCs may show considerable variations. Furthermore, SCM is, undeniably, one of the key success factors for software organisations. In a changing environment, the SCM process needs to be developed further and revised to better support the new ways of developing software. What this thesis aims at is to convey some significant parts of this important area. From this perspective, we are able to derive the following research problem:

How can the SCM process be analysed, when it takes place within the distinct context of a VSC environment, so as to provide a basis for improvement?

The SCM process analysis and improvement approach developed within this research is based on the work done in the context of the VISCOUNT (*Virtual Software Corporation Universal Testbed*) project. The project has been set up to increase the understanding of the SCM process within VSCs, and to develop methods and tools that would support industrial organisations in meeting the new SCM challenges. Within the application organisations of the VISCOUNT project, debis Systemhaus GEI (debis), Neles Automation (Neles), and Societa Italiana Avionica (SIA), improving the SCM process has been seen as one of the most important means of increasing the efficiency of the software process (McGowan 1997). Another framework for the dissertation has been provided by Pr²imer (*Practical Process Improvement for Embedded Real-Time Software*), an SPI action cycle model, which has been applied and further developed at VTT Electronics (Karjalainen et al. 1996, Komi-Sirviö et al. 1998). The main contribution of the thesis is the establishment of a new approach to analysing and improving SCM within a VSC.

As already stated above, the analysis of the current status of the process establishes a baseline for the process improvement work. The analysis cannot be carried out properly unless the contents of the process are identified. No distinct definitions for the SCM process when taking place within a virtual environment have been published so far. A specification of the SCM requirements is required for defining the SCM process for VSCs. In other words, the grounds for the enlargement must be asserted. For this reason, a systematic method of defining the SCM requirements is presented in this dissertation. The expanded SCM process definition introduced in the thesis is based on the industrial SCM requirements collected in connection with the research and the literature survey. Finally, we present how the SCM requirements analysis approach and the expanded SCM process description can be utilised when developing the SCM process to better meet the defined VSC-specific requirements.

On the basis of the discussion above we can formulate the following questions to solve the general problem:

- Q1 How can the SCM requirements be defined for a VSC?**
- Q2 How can the SCM process be expanded for a VSC?**
- Q3 How can the SCM process be developed to meet the VSC requirements?**

1.3 Research Strategy

Software Engineering encompasses procedures, methods, and tools that allow controlling the process of software development, and provide a foundation for software professionals to produce high quality software in a cost-effective way (Pressman 1993).

Regardless of the practical purpose of software engineering, researchers have been heavily criticised for their inability to influence industrial practices and for the quality of the resulting software (Potts 1993, Glass 1994, Parnas 1998). These accusations emphasise the fact that software engineering research should be based on the problems and requirements of industrial organisations, and that the research process should produce solutions to these practical problems. This problem-driven approach relies on gaining a thorough understanding of the application environment, and on the interaction between the technical and non-technical factors of the software process (Potts 1993). Case study research emphasising the analysis of real-life examples of software development organisations has been regarded as a viable method of producing useful research results. The research strategy of this dissertation is based on the case study research method (Yin 1991). Here, the research is divided into the following four tasks:

- 1) observe the existing problems and solutions regarding SCM,
- 2) analyse the SCM requirements in a VSC,
- 3) analyse and explore a proposal of developing SCM to meet VSC requirements,
- 4) implementation, evaluation, and validation of the proposal.

This division is based on the research model presented in (Glass 1995). The focus of this thesis in the context of research tasks lies on the matters presented in the following sections.

Observe the existing problems and solutions regarding SCM. The main purpose of the observation task is to define and to analyse the current SCM problems and solutions within the application organisations. This approach is based on understanding the case environment. In other words, the general characteristics of virtual corporations are first examined and later analysed against the description of the case environment. The information gathering during this stage is based on various material sources, e.g. VISCOUNT project documentation, interviews, and literature survey. Thus, through close interaction with several industrial organisations, it has been possible to identify the current state and initial requirements of the SCM process. The results have established the base for the following tasks by providing an understanding of the SCM challenges within the application organisations. Hence, the research is grounded on real-world problems and needs. In paper I, the practical experiences from the current state analysis made within one of the application organisations are described.

Analyse the SCM requirements in a VSC. The goal of the second research task is to analyse the SCM requirements within the application organisations. During the task, a practical approach for defining SCM requirements has been applied. The requirements analysis approach, as well as the initial results and experiences are described in papers II and IV.

Analyse and explore a proposal of developing SCM to meet VSC requirements. The results of the previous tasks are exploited, in addition to the experiences from the application organisations and results of the literature survey, for identifying solutions enabling the SCM process to be developed towards a virtual process. Furthermore, the identified requirements and the expanded SCM process description are used as a basis for a VSC-specific SCM process improvement framework, this being one of the main results of this work. The framework can be used when developing SCM within VSCs. The practical experiences gained with the SCM improvement cycle within an application organisation are presented in papers I, II, III, and V.

Implementation, evaluation, and validation of the proposal. During the last research task, the solutions are implemented, evaluated, and validated by means of experimentation and observation. As presented in Chapter 5 of the dissertation, the SCM analysis and development approach for VSCs is the result of combining the experiences from the application organisations with the outcome of the literature survey. This approach involves both tested and derived solutions, presented in such a form that makes it possible to exploit them within other virtual software corporations as well. In Chapter 6, the main results of the dissertation are evaluated and validated. As a context for assessing the quality of the case study research results we employ the four tests presented in (Yin 1991): construct validity, internal validity, external validity, and reliability.

1.4 Outline of the Thesis and Main Results

This thesis deals with software configuration management (SCM) within virtual software corporations (VSC). Emphasis is placed especially on increasing the understanding of the SCM process within a VSC, and on the development of a systematic approach for analysing and improving SCM so as to meet the VSC-specific requirements. The structure of this dissertation is presented in the following:

Chapter 1 gives an introduction to the subject and justification for the research problem. Furthermore, the research strategy applied in this research is presented.

Chapter 2 introduces the related work of this dissertation from the viewpoint of software configuration management (SCM) and virtual software corporations (VSCs). In addition to the basic concepts and traditional definitions of SCM, the general characteristics of VSCs are presented. The challenges of SCM within VSCs are also discussed.

In Chapter 3, the case environment of this dissertation and a systematic method for defining SCM requirements within VSCs are presented. The method is based on the widely accepted use case approach. Thus an answer is attempted to the first research question, “*How can the SCM requirements be defined for a VSC?*”, by presenting a case example of SCM requirements analysis within a VSC. As a result of this research, the justifications for creating an expanded

SCM process description are proposed. Papers II and IV present the case experiences related to this chapter.

In Chapter 4, the traditional SCM process definition is expanded to cover SCM teamwork, improvement, and planning sub-processes. Thus, chapter 4 establishes the SCM sub-processes of a VSC, aiming at answering the second research question, “*How can the SCM process be expanded for a VSC?*”. As such, the description is primarily intended as a tool for SCM process improvement teams. Consequently, the structured and expanded SCM process description can be considered an instrument for reviewing the current status of the SCM process and for revising it. On the other hand, the description can be used as a starting point for designing or tailoring the SCM process for a VSC. Consequently, the expanded SCM process description forms the basis for the SCM development framework presented in the following chapter.

Chapter 5 introduces an SCM process improvement framework for VSCs. A software process action cycle model called Pr²imer (*Practical Process Improvement for Embedded Real-Time Software*) is used for supporting the SCM development within VSCs. The Pr²imer model has been successfully applied in several industrial SPI projects at VTT Electronics. Here, the general framework is focused towards SCM and, consequently, a practical tool is introduced for analysing the SCM process and developing it into a virtual SCM process. The case examples as well as the experiences gained with the improvement cycle are presented in the accompanying papers I, II, II, and V.

In Chapter 6, the research results are examined and validated with regard to the scope of the study and the methods used. The expanded SCM process description has been created mainly on the basis of the SCM requirements identified during our study. First of all, it is evident that the virtual character of the business structure does affect the SCM process and that the major changes concern teamwork activities. However, as the research relies heavily on the case study research strategy, the representativeness of the description may be brought into question. In this chapter, this issue is discussed, and the expanded SCM process description is validated, as is also the SCM-Pr²imer approach for VSCs.

Chapter 7 presents the final summary of the research, its results, and the limitations of the study. In addition, some related future research topics are discussed.

Chapter 8 provides an introduction to the original papers I–VI, included as appendices.

2. Related Work

2.1 Introduction

This dissertation deals with the challenges to software configuration management (SCM) met by software companies when confronted with certain types of business process effects. The focus is, to be precise, on virtual software corporations (VSCs), rather than on software companies in general. VSCs can be described as multi-organisations operating within distributed, dynamically changing, virtual environments. When meeting these new challenges facing SCM in virtual organisations, the first step is to understand that the increasing complexity and number of business processes will result in changes which cannot be solved by means of traditional SCM procedures. This implies that the efficiency of many SCM principles and techniques now widely used will be compromised when an organisation is changing towards a more virtual structure and distributed resources. Furthermore, the basic changes are related to the SCM process definitions.

The main objective of this dissertation is to examine how SCM practices could be analysed and turned towards a virtual process. The key references related to the research problem are the SCM process definitions presented in the literature. Several models have been published in various standards and software process assessment approaches. In addition to the models, the related literature also provides a number of research and case reports concerning software process improvement. Although the SPI research is closely related to this dissertation, it is not the main research topic here. The main reason for not incorporating this issue is the fact that an extensive research on the topic has been published (Kinnula 1999). In addition, SPI experiment reports rarely focus on the SCM process (Cugola et al. 1997), and even if they do, the scope is typically limited to the traditional single site development or to SCM tool experiences in a single organisation (e.g. Auer & Taramaa 1996, Larsen & Roald 1998, Crnkovic & Willför 1998). Virtual organisations have been covered in various publications, exploring the subject from a diversity of viewpoints. General VSC attributes and characteristics presented in the literature form a basis for studying the key references related to this work.

In order to establish a theoretical depth of the results of this dissertation, the scope of their usefulness and applicability to software companies in general, the foundation of SCM must be clarified first. Accordingly, this chapter begins with a summary of the basic concepts connected with SCM (section 2.2). The most widely known SCM process definitions are briefly introduced in section 2.3. The purpose is to point out the basic structures, which are then used as a basis for developing an expanded SCM process model for VSCs. A survey of the virtual organisation literature provides a basis for the discussion concerning SCM challenges within VSCs (see section 2.4). In section 2.5, the related work is summarised and analysed with regard to the objectives of this dissertation.

2.2 Basic Concepts of Software Configuration Management (SCM)

In this section, the basic SCM-related concepts are introduced briefly. The reason for the conciseness is that these concepts are presented at length in several books and publications (e.g. Tichy 1988, Compton & Conner 1994, Buckley 1996). On the other hand, since these concepts are vital for understanding the basic issues dealt with in this dissertation, a brief presentation of them is included to provide a conceptual background for the reader.

Software Configuration Management (SCM) is a discipline for establishing and maintaining the integrity of the products of a software project throughout the project's lifecycle (Tichy 1988). The basic element of SCM is called *Software Configuration Item (SCI)*, which can be a unit or a collection of lower-level items. SCIs include items such as source code, executable code, documentation, test data, designs, specifications and project management information.

During software development and maintenance, an SCI undergoes a series of version changes (Compton & Conner 1994). A *version* is a state of SCI that can be defined as either a variant or a revision. *Variants* indicate divergence in sets of versions that are intended to coexist. Variants are made due to some physical or functional aspect, such as different platforms. *Revisions* are SCIs that change with time and may be created for various reasons, e.g. for fixing bugs or enhancing functionality (Conradi & Westfechtel 1996).

A specific version used as a reference point for further development is called baseline. The *baseline* is a tested and formally accepted version of a software product. A version becomes a baseline through a change control procedure, i.e. when a group responsible for the task decides to designate it as such (Compton & Conner 1994, IEEE-610.12 1990, IEEE-828 1990). A *release* is a particular version of an SCI that is made available for a specific purpose, e.g. for testing or customer delivery (ISO/IEC-12207 1995).

2.3 SCM Process Definitions

As stated above, when analysing and improving a software process, it is vital to understand the content of the process (Paulk et al. 1995, Jones et al. 1999). The SCM process has been defined and classified in various standards and process models, aiming at describing the regular expectations of an SCM process, irrespective of how the process is implemented in an organisation. In other words, the models and standards describe the basic elements and activities involved in the process. In this section, the usual definitions are identified and presented as a short summary in view of the SCM process definition. There are two reasons for the introduction; the definitions have become established in SCM research; and thus, they also form a basis for analysing and improving the SCM process also in a VSC environment.

The most common descriptions regarding SCM are presented in IEEE standards (IEEE-1042 1987, IEEE-828 1990, IEEE-610.12 1990). These standards divide the SCM process into four main functions: configuration identification, configuration control, status accounting, and configuration audits and reviews. Configuration identification refers to identifying, naming, and describing the documented physical and functional characteristics of the code, specifications, design, and other data elements to be controlled in the project. Configuration control has to do with requesting, evaluating, approving or disapproving, and implementing changes. The objective of status accounting is to report the status of the project configuration items, while configuration audit is used for determining to what extent the actual configuration item reflects the required physical and functional characteristics. The content of the SCM plan is also described in IEEE standards.

SEI, the Software Engineering Institute (CMU/SEI-92-TR-8 1992) broadens the categories of IEEE by supplementing the definition above with manufacturing, process management, and teamwork functions. The goal of manufacturing is to manage the construction and building of the product, whereas process management aims at ensuring the correct execution of the organisation's procedures, policies, and life-cycle model. According to the standard, the teamwork element deals with the concepts of workspace, transparency and transactions. These elements are thus related, from a technical viewpoint, to the co-ordination of software engineering teams.

ISO, the International Organization for Standardization, divides the configuration management process, according to the ISO 12207 standard (ISO/IEC-12207 1995), into the following activities: process implementation, configuration identification, configuration control, configuration status accounting, configuration evaluation, release management and delivery. Process implementation refers to the planning of configuration management activities. The contents of the other processes correspond, in general outline, to the IEEE descriptions.

The purpose of the Capability Maturity Model for Software (CMM) is to “guide organisations in selecting process improvement strategies by determining their current process maturity and identifying the few issues most critical to improving their software process and software quality” (CMU/SEI-94-HB-04 1994). The model has five process maturity levels, each of them including specific key process areas. SCM is one of the key process areas at the maturity level Repeatable. Each of the key process areas is described in terms of key practices, which describe the activities of the process area. SCM activities can be incorporated to the SCM process definitions described above.

The Bootstrap method (Kuvaja et al. 1994) for software process assessment and improvement divides software processes into three process area categories further divided into process areas. The Methodology category of Bootstrap includes the support process area. The support process area is divided into processes, one of which is the configuration management process. The process consists of a set of best practices, which can be incorporated into the definitions IEEE and ISO 12207 (see above).

The forthcoming ISO 15504 standard (ISO/IEC 15504-2 1998) groups software processes into primary, organisational, and supporting lifecycle processes. The groups are divided into five process categories, one of which is the Support process category. The Support category contains the configuration management process, divided into six basic processes. The basic processes are aligned with those defined in the ISO 12207 standard.

As a conclusion, according to the traditional definitions, SCM consists of the following processes: SCM planning, configuration identification, configuration status accounting, change control, configuration auditing, as well as release manufacturing and management. In Figure 1, the SCM base processes are presented in one entity box. SCM planning has often been defined as a process to be used for planning, monitoring, and controlling the base processes, as well as in the procedures of ensuring that the software is developed according to a specified software development methodology (e.g. ISO/IEC-12207 1995). This relationship between the SCM planning process and the base processes is illustrated in Figure 1 with the arrow between the elements.

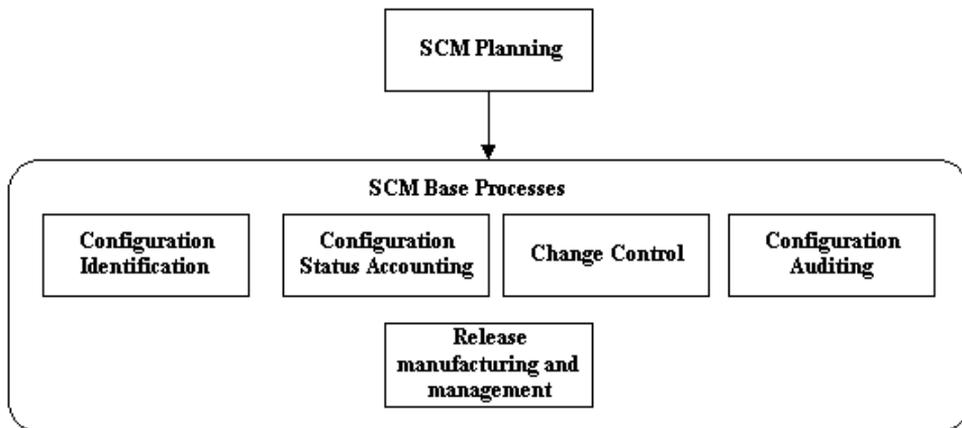


Figure 1. The traditional definition of the SCM process.

Our case study demonstrates, however, that the traditional SCM process definition is not sufficient for analysing and improving the SCM process within a VSC. On the contrary, it is apparent that VSC-specific characteristics do affect the SCM process, therefore making changes also to the process definition necessary. In the following sections, the issues of what the general attributes and

characteristics of a VSC are, and how they affect the SCM process, are discussed both in general and specifically concerning the application organisations of the VISCOUNT project.

2.4 Virtual Software Corporations and SCM

2.4.1 Introduction to Virtual Corporations

In addition to *virtual corporation* (Davidow & Malone 1992), there are several other names for similar organisational structures, such as *network organisation*, *cluster organisation*, etc. (Alstyn van 1997). Virtual corporation is also often associated with other seemingly similar phenomena, namely *virtual memory*, *virtual reality*, and *virtual offices* (Mowshowitz 1997). The different content and emphasis of virtual corporations are closely connected with the term *virtuality*, in particular when compared with other organisational structures. Virtual corporations are temporary, dynamically changing networks of companies, whose employees are dispersed throughout an interconnected world of information systems. Moreover, the virtualness can be seen as a strategic characteristic within every organisation (Venkatraman & Henderson 1998). In other words, the concept is applicable to various types of multilateral activities carried out in various organisations. In conclusion, it can be stated that even though a virtual corporation, consisting of a web of firms, individuals, technologies, and assets, may have a vague physical identity, it still has an organisational structure.

The virtual corporation is a teamwork environment where all the members work together in order to reach a common goal (Davidow & Malone 1992, Lipnack & Stamps 1997, Ahuja & Carley 1998). Ideally, the expertise of the team members combined with efficient utilisation of tacit knowledge creates the key success factor (Nonaka & Takeuchi 1995). This requires a high degree of both informal and formal communication between the teams (Ahuja & Carley 1998), as well as information sharing and integration across formal boundaries. People and other resources need to be efficiently interconnected, and therefore appropriate information technology support plays a major role in virtual corporations.

The virtual corporation seems to be a far more dynamic and flexible than any other organisational form. When it comes to reacting quickly to specific market changes and opportunities, the strength of the virtual corporation lies in its ability to exploit knowledge and skills flexibly and dynamically throughout its collaboration network (Alstyne van 1997). In this regard, the virtual corporation also presents specific opportunities and challenges to customer interaction, offering a “two-way information link between a company and its customers” (Venkatraman & Henderson 1998). The existing business web and information technology solutions offer customers easy access to products and services, regardless of the location of the product or supplier. In such a scenario, customers are able to participate in dynamic product customisation i.e. they can take part in some of the activities in the SCM process (Venkatraman & Henderson 1998).

2.4.2 VSC Attributes and Characteristics

Organisation structures have been viewed from various perspectives. For instance, Venkatraman and Henderson (1998) have studied organisations, and defined virtualness as a strategy reflecting three vectors: customer interaction, asset configuration, and knowledge leverage. There is a whole scientific community of marketing researchers who have studied networks of collaborative industrial relationships (Håkansson & Snehota 1995). Marshall van Alstyne (1997) has studied network organisations from economic, social, and computational points of view. In our research, we will concentrate on virtual software corporations from the viewpoint provided by SCM. Alstyne’s computational metaphor, which “draws attention to design, parameters for tasks, processors (or managers), their arrangement, and communication among them” thus relates to our work. However, when analysing and improving SCM concepts, the other points of view should not be totally abandoned. For example, the original needs of the application organisations of the VISCOUNT project are quite closely combined with cost-efficiency and project management, as well as corporate culture and other non-economic aspects of human nature (McGowan 1997). Overall, there is a great number of publications and articles focusing on virtual organisations, some of them dealing especially with SCM (e.g. Boldyreff et al. 1996, Hoek van der et al. 1996, Van De Vanter & Murer 1999). Even though the viewpoints of the research do vary, a mutual understanding on two distinctive VSC-specific features is emerging. These features, or *VSC attributes*

as they are called in this dissertation, are virtual structure and virtual resources. They are also presented briefly in paper IV, and later developed as a basis for a survey of literature and for the examination of the organisational and project models in the application organisations.

Virtual Structure. One of the main features separating a VSC from other types of software production organisation is the structure. In sharp contrast to a single site software development, the VSC is characterised by *distribution*. In other words, while many organisations have been seen as a single unit operating at a single site, the teams of a VSC are typically distributed in different locations. Different organisations often show various types of distribution (Asklund et al. 1999). For instance, while a VSC may have two teams operating in a single city, another organisation may have several teams and experts working at a number of locations around the world. Furthermore, the teams and individuals involved may be working simultaneously for several companies or projects, sharing a common interest or goal only when working for a specific VSC.

Impermanence is an important characteristic of the virtual structure (e.g. Davidow & Malone 1992, Mowshowitz 1997). Virtual structure involves, for example, inter-firm alliances and the use of sub-contractor networks. Immediate needs for labour in a VSC are more likely to be met through short-term contracts rather than permanent agreements (Venkatraman & Henderson 1998). Consequently, the bonds between the VSC teams can be of temporary nature, and often related to a specific project. However, it should be noted that VSC partners are most likely to aim at ensuring the possibilities for a longer-term strategic collaboration, e.g. through managing portfolios of relationships (Ford et al. 1998). Therefore, as is the case with the distribution characteristic, a VSC may also have several levels of impermanence.

Even though various characteristics for virtual structure can be found in the literature, we will view the virtual structure in the context of distribution and impermanence. There are two reasons for the brevity; distribution and impermanence seem to be the most common characteristics in current VSC research; in addition, they are also the most important aspects of the application organisations of the VISCOUNT project (McGowan 1997). The following three examples from the application organisations are presented to illustrate the challenges provided to SCM by the virtual structure.

- Debis Systemhaus, Germany (debis), a company of DaimlerChrysler Services, known internationally as debis IT Services, is one of Europe's leading information technology service providers. Debis employs people in 20 countries. The main market sectors are manufacturing industry, financial services, telecommunications, public sector, retail, as well as travel and transport. Debis assembles the required capabilities for software projects through a coalition, including production at the various sites of the company and the use of subcontractors. Thus, the strategic advantage of debis seems to be based on the superior organisational structure. Creating an SCM *infrastructure* for a coalition and navigating in a fast-changing environment is not a simple task. The persons in charge of the software projects at debis are especially concerned with the basic issues concerning how to build a functional network architecture for SCM, and what the most practical SCM tool sets for the teams are.

- Neles Automation, Finland (Neles) provides solutions for production and quality management in pulp, paper and energy industries. As a part of an attempt to create an effective business model, Neles is, in some software projects, relying on external sources, not only for support activities (e.g. SCM tool support), but also for critical resources. Under these circumstances, the SCM *facility* has proved to be one of the most crucial challenges of the projects, although Neles is already employing an SCM tool designed to be applied within a geographically distributed environment. Within such an environment, Neles has faced special challenges caused, for instance, by concurrent software changes, data transmission between different sites, and database administration. In addition, it has become evident that commercial SCM tools must be better integrated with each other and with project management tools. This requirement is relevant due to the fact that it is not always possible or economically reasonable to use the same tools. On the other hand, obstacles may arise from the fact that a company may have used a certain tool for a long time and is therefore unwilling to change for a new one.

- Società Italiana Avionica, Italy (SIA) develops software for aeronautical, spatial, naval, and defence applications. The software projects of SIA are typically distributed to different sites and involve subcontractors as well. Since most of the company's products are classified, the *security*

requirements for SCM are tight. The security aspect can be considered from the distribution and impermanence viewpoints. Since the applications involve large percentages of data being transmitted over distributed teams, the VCS should be able to support security issues, such as encryption, compatibility with corporate firewalls, and restriction of access for specific types of transmission. While a VSC structure might be temporary, the rules and the procedures for the security levels within a virtual project should be made more consistent, pre-planned and uniformly implemented. The critical questions for SIA include the following: What information and data a subcontractor can or should have at its disposal? What are the methods and techniques for documentation security? Who has the right to make changes to a specific feature? How to formulate the privacy and access control policy? And how to distribute data safely and as resiliently as possible? These questions are examples of the everyday security challenges related to the virtual structure of SIA.

Virtual Resources. The term virtual resource is used here to denote the VSC characteristics identifying and describing elements less concrete and visible than distribution and impermanence discussed above. Virtual resources include teams and dynamism. *Teams* is an issue that has been under discussion in several publications and books in the past few years. The issue has also been regarded as one of the basic elements of VSCs (Davidow & Malone 1992, Lipnack & Stamps 1997, Ahuja & Carley 1998). Undoubtedly, the team concept can be studied and analysed from multiple viewpoints. We have developed our logic for teams by placing SCM in the centre. In other words, we focus on SCM methods and techniques that support the software production teams within a distributed and non-permanent environment.

The accelerating pace of change in industry results in an increase of *dynamism* in business operations. Consequently, an ever-greater share of operations is based on spontaneity and ad-hoc decisions, sharp wits and quick reactions (Ståhle & Grönroos 1999). Ideally, dynamic organisations can react quickly to the challenges of competitors and changes in the business environment, thus being capable of managing the rate of change. To support dynamism, a company should pay attention to the flexibility of the SCM process, as well as to the adaptability and customisation of SCM methods, tools and techniques (Noll & Scacchi 1997).

Looking at the VISCOUNT application organisations, it is evident that virtual resources make the SCM process more complex, which again presents new challenges to the organisations. Single software development teams do not, for instance, face any *cultural* differences related to the use of different SCM methods, tools, and techniques (Noll & Scacchi 1997, Van De Vanter & Murer 1999). Moreover, the *communication* between the teams involved is much more flexible and reliable when working in a single site as opposed to managing a project over organisational, linguistic, and methodological boundaries. Furthermore, the requirements for dynamism within VSCs are likely to bring about more complexity, as organisational structures, teams, processes, and the application content are not as stable as they typically are in a single-site development environment. In other words, a VSC should be capable of *switching* resources and borrowing know-how and other capacities when necessary, thus optimising its organisational performance (Alstynne van 1997). Since the switching of resources, be they human, process-related, or technical, makes it possible to improve resource utilisation, the provision of SCM support should be ensured.

The following scenario illustrates the case. Collaboration among two organisations is formed to enhance a new telecommunication system. The organisations have a lot of contenders in the application area, while the fast changing markets causes many changes to customer requirements. The project involves software teams at three different sites, one operating in Germany, one in Russia, and one in the USA. In addition, one of the organisations has hired subcontractors for producing some crucial parts of the products while the project is already in progress. Even though the new experts would have the time to work on the project, a central part of the product can not, without extra effort, be detached from the main repository to achieve the expected results. Furthermore, the teams from the two organisations each have their own specific SCM tools and follow somewhat customised SCM practices. The tools do not have any automatic interfaces, while a lot of data should be transferred securely from one site to another. This kind of scenario would, without any doubt, cause considerable problems to project management and software release manufacturing.

2.4.3 Summary of Virtual Software Corporations and SCM

The VSC attributes and characteristics discussed above and summarised in Figure 2 present challenges to SCM, visible not only within the VISCOUNT application organisations but also, to some extent, in recent literature (e.g. Hoek van der et al. 1996, Noll & Scacchi 1997, Estublier 1999). However, as virtual organisations may in practice differ a lot from each other, virtualness appears in various ways. Furthermore, some organisations may display only few of the VSC-specific characteristics, or are just developing their business processes towards virtualness. Hence, careful preliminary studies are needed at the beginning of any SCM improvement program involving distributed organisations.

To provide a basis for such studies, we have specified two attributes and four characteristics for VSCs, corresponding to the key SCM challenges. Our approach recognises the VSC attributes of virtual structure and virtual resources, as shown in Figure 2. Virtual structure refers to the VSC characteristics of distribution and impermanence. These characteristics may provide a powerful concept for enhancing the efficiency of an organisation. However, they also create new challenges related to the infrastructure, facility and security issues of SCM, in particular. Virtual resources may be human, process-related, or technical assets created and utilised by the organisation. Here, the teams and the characteristics of dynamism are used for explaining the new challenges facing SCM that have to do with culture, communication, and resource utilisation.

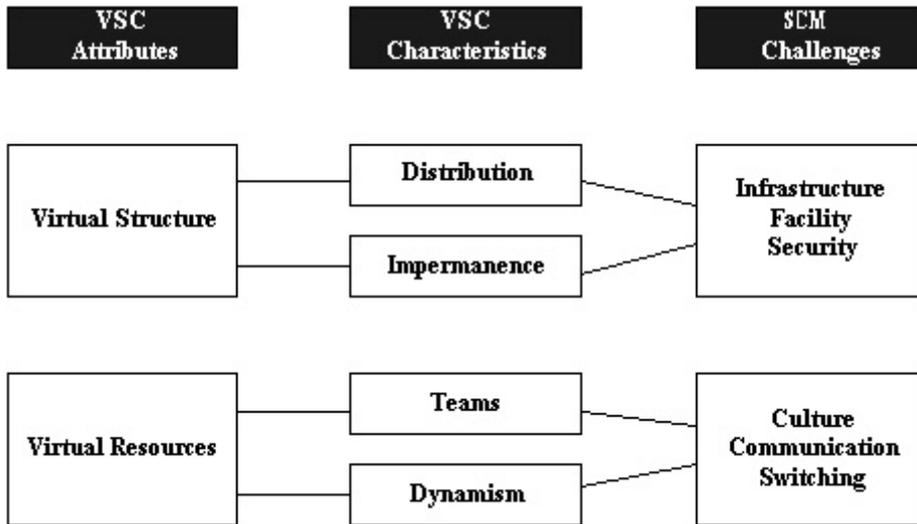


Figure 2. VSC Attributes and Characteristics vs. SCM Challenges.

2.5 Summary of Related Work

Although there are several related articles that can be used as sources and references in defining and improving the SCM process in general, they all show limitations when focusing on SCM process development in the specific context of VSCs. Van De Vanter and Murer (1999) refer to this issue as *organisational complexity*. In this dissertation, this particular dimension has been investigated by means of VSC-specific attributes and characteristics, which have been identified by examining the related literature and the application organisations of the study. The key lesson is that the increasing complexity of organisations will result in changes in the SCM process as well. It is evident that the SCM-related challenges in a VSC are different from those in a traditional single-site software organisation due to the specific VSC characteristics. The focal characteristics of our study are distribution, impermanence, teams, and dynamism. The traditional SCM definitions describe SCM base processes in broad outline, failing to pay enough attention to the expanded SCM view.

All the most widely known process assessment and improvement methods include the SCM process as one of the basic software processes. However, these methods focus largely on the whole organisation and are likely to limit their

view to analysing and developing only a single target process. Although these common methods allow SCM-specific process analysis to be carried out, they do not support the expanded view that would be required for VSCs, since their idea of SCM is based on traditional process definitions.

There are numerous publications available on virtual organisations. The VSC-specific characteristics, described in the previous sections, form a framework for analysing a virtual organisation. In other words, the characteristics are not necessarily typical in all VSC organisations, or they may exist only to some extent. Therefore, there is a strong case for developing a method of extending and specialising the general characteristics so as to improve their adaptability and facilitate their use when employed in an SCM improvement program.

The need for a systematic approach to SCM improvement towards a virtual process may, however, not be readily apparent in an organisation. On the other hand, the rapidly emerging VSC attributes and characteristics are so striking that the persons involved will soon realise that a new view to SCM process analysis and improvement is needed (e.g. Noll & Scacchi 1997, Asklund et al. 1999, Frühauf & Zeller 1999, Van De Vanter & Murer 1999). Furthermore, looking at our application organisations, it is evident that a new dimension to SCM process analysis and improvement is also required (McGowan 1997). The importance of a virtual SCM process became apparent in the application organisations of the VISCOUNT project. The following chapter introduces an SCM requirements analysis for a VSC. The results of the analysis within the application organisations of the VISCOUNT project prove that SCM is one of the most crucial factors in virtual software organisations.

3. SCM Requirements Analysis in a VSC

3.1 Introduction

The first step towards a successful implementation and use of software development tools and practices in an organisation is to understand the processes that they need to support (Wein et al. 1995). The main purpose of this dissertation is to investigate how the SCM process can be analysed, when it takes place within the distinct context of a VSC environment, so as to provide a basis for improvement. The analysis cannot be done properly if the activities of the SCM process have not been identified. The related work as well an initial analysis of the application organisations showed that the current SCM process definitions are not sufficiently extensive when dealing with virtual organisations. Hence, there is a clear need to analyse the SCM process within a VSC environment, so as to be able to define the SCM requirements, and thus also the expanded content of the SCM process.

The second section of this chapter establishes the organisational context of the case. In the third section, the analysis approach, based on the use case technique, is introduced. Here, the first research question, "*How can the SCM requirements be defined for a VSC?*" is answered by presenting a case example of SCM requirements analysis, together with our analysis of it and the results obtained. The results show that in the case of our application organisations, a sound understanding of the virtual SCM process is of vital importance to identifying the requirements, thus also implying a need for more comprehensive user requirements. Furthermore, our application organisations each show their unique combinations of processes, strengths, and weaknesses; similarly, the goals of SCM development are varied. A systematic requirements analysis method supporting each organisation on a one-to-one basis is therefore necessary. The initial analysis and results of the requirement analysis in the application organisations are presented in papers II and IV.

3.2 Case Environment

3.2.1 VISCOUNT project

This section introduces the VISCOUNT project, which establishes the context of this dissertation. The main source of this section is the project plan (McGowan 1997), its sub-parts, and <http://www.viscount.org> (public web page), as of January 2000.

VISCOUNT (*Virtual Software Corporation Universal Testbed*) is an EU/Esprit project, project code #25754, carried out during 1997–2000, comprising the following partners:

- ❑ British Aerospace Ltd., UK (BA),
- ❑ debis Systemhaus GEI, Germany (debis),
- ❑ Glasgow Caledonian University, Scotland (CGU),
- ❑ Neles Automation, Finland (Neles)
- ❑ Società Italiana Avionica, Italy (SIA),
- ❑ VTT Electronics, Finland (VTT).

One of the main goals of the VISCOUNT project has been to further develop LIFESPAN™, a configuration management tool to be used in a virtual software corporation. The other goals of the project are (McGowan 1997):

- ❑ to create greater awareness for all the issues surrounding software development in VSCs, with a particular emphasis on configuration management,
- ❑ to increase the awareness of the potential for tool and infrastructure support for VSCs, and
- ❑ to gain a better understanding of the SCM processes necessary for supporting software development activities in a VSC, and to measure their efficiency.

The goals of this work are mainly related to the first and the last of the VISCOUNT project goals presented above, and the author of this thesis has

actively been participating in those tasks. The project has proceeded in three stages, each of them having their own viewpoint. In the first stage, the current status of SCM has been analysed within the project's application organisations. The focus of the second stage has been on distribution. In other words, the aim has been to analyse what SCM-related challenges and problems are caused by the distribution of work into different sites. Moreover, the other VSC characteristics, such as impermanence, teams, and dynamism have been evaluated in the context of SCM. The third stage has been concerned with process modelling and metrics. The purpose has been to analyse how the defined properties could be used to support virtual organisations. In each stage the procedure comprised the following steps:

1. Clarifying the requirements by considering both user requirements, i.e. the requirements of the application organisations, and technological recommendations.
2. Carrying out detailed design and planning work, including LIFESPAN™ development, along with the plans for SCM process development, evaluation, and testing. The testing and evaluation plans included both LIFESPAN™ user trials and trials of improved SCM practices.
3. Developing a working LIFESPAN™ prototype, and defining SCM guidelines, instructions, procedures, and metrics collection support.
4. Evaluating the prototype and the SCM practices in a real-world environment and evaluating development success according to the evaluation plans.

The author of this thesis has been involved in all the main tasks mentioned above, except LIFESPAN™ development, i.e. the third task. In other words, the research, the results of which are described in this dissertation, has been carried out while working on the other main tasks. On the whole, the participants of the VISCOUNT project have had different roles in the project. British Aerospace Ltd. (BA), for instance, has been responsible for developing the LIFESPAN™ tool and giving technological support to the other partners. The three application organisations, debis, Neles, and SIA have, for their part, provided an industrial testbed for the LIFESPAN™ prototype. Moreover, the application organisations have been interested in analysing their current SCM practices and developing

their SCM processes, aiming at these providing a better support for their methods of doing business. Glasgow Caledonian University (GCU) has been responsible for examining the technical possibilities for virtual SCM, studying, e.g., different SCM tools and distribution mechanisms.

The role of VTT Electronics (VTT) and thus also that of the author of this thesis have been to research the SCM process and its development towards a virtual SCM concept. The main task has been to provide the application organisations the methods and the framework needed for analysing and defining the SCM, identifying the requirements, and developing the process towards virtual SCM. In addition, the author of this thesis has been responsible for collecting, evaluating, and analysing the results. In connection with the research project, the author has also spent some time working at Neles, one of the application organisations. The author of this thesis has been the person bearing the principal responsibility for carrying out the tasks assigned to VTT. Thus, along with presenting and elaborating the research ideas of this dissertation, the author has been responsible for carrying out most of the practical work required for the development of methods and framework, as well as the evaluation and analysis of the results.

3.2.2 Application Organisations

This section presents the application organisations of the VISCOUNT project. Describing the organisations in case study research is of great importance, as the representativeness of the results has to be validated. In our study, this means that it has to be shown that the application organisations can be considered a sufficiently typical and comprehensive representation of virtual organisations. In addition, while the research results are generalised from a single organisation to an organisation type, it is unreasonable to expect generalisation to all organisations. Thus, the results of the research must be considered relative to a specific range of problems or development context (Potts 1993).

The application organisations studied in this thesis are debis Systemhaus (debis), Neles Automation (Neles), and Società Italiana Avionica (SIA). Within these industrial organisations, SCM has been seen as one of the most important means of increasing the efficiency and effectiveness of the software process (McGowan 1997). The main information sources for this section are the public web pages of

the organisations (<http://www.debis.de/debis/systemhaus/>, <http://www.neles-automation.com>, and <http://www.sia-av.it>), as of January 2000. Furthermore, as the author of this dissertation has been working for the VISCOUNT project from the beginning, the information was supplemented by the author providing her personal knowledge and experience of the organisations.

The annual turnover of **debis Systemhaus (debis)** amounted to EUR 2.1 billion in 1999. debis offers a comprehensive portfolio including consulting services, software solutions, desktop services, as well as the management of applications, data centres and communications networks. With more than 16.000 employees, debis Systemhaus is the largest manufacturer-independent service provider in Germany.

For software projects, the required capabilities are assembled through a coalition, including production in the different sites of the company, subcontractors and individual experts. The profiles of the various projects may differ significantly from each other. Some are carried out in demanding, dynamic environments where competition is extremely hard and customer requirements are changing fast. Thus, managing the rate of change in information-intensive markets and projects presents a great challenge to debis. SCM is considered one of the major critical areas in current software development work at debis. Several problems have been identified that are directly or indirectly related to the dynamic, temporary teamwork environments. For instance, according to the debis experience, virtual environment is likely to significantly reduce the availability of SCM information to project members, due to the distance between the teams, the lack of common tools and procedures, and the absence of a single database. This situation affects the management capabilities of projects, making it harder to optimise project tasks, to manage available resources, and to control the ongoing project. This will increase the danger of wrong decisions due to misinformation and lead to additional risks with respect to project quality and progress. Therefore, the company is running a continual research on how to assist virtual projects and how to optimise the processes used in them.

Neles Automation (Neles), a company belonging to the METSO corporation, is a combination of Valmet Automation and Neles Controls, the suppliers of process automation and flow control solutions. In 1998 the net sales of Neles

totalled EUR 600 million and the number of employees amounted to 4500. Neles develops supplies and provides service for extensive measurement and control solutions, as well as control valves and digital control systems. These systems are used for measurement, analysis, and monitoring, automatic control and process information management tasks at pulp and paper mills, power and desulphurisation plants, on oil and gas pipelines and in other industrial processes. Customer services, sales, and maintenance are carried out via a global network operating in 30 countries. Within this environment, Neles competes on an open, international, and very demanding market. The case experiences of this study have been collected from the Valmet Automation part of the organisation. Therefore, the experiences are not necessary typical of Neles as a whole. However, as the companies have now been combined, the structure of the organisation has become larger and more elaborate. Consequently, the challenges related to the SCM are more likely to increase rather than decrease.

The types of software projects within Neles are likely to vary considerably. In addition to several small maintenance projects, the company is running some large pioneer projects applying new technology and solutions. The tendency in the software projects is towards distribution and resource coalitions, and thus a more expanded use of sub-contractors and individual experts. However, Neles has now reduced the speed of decentralisation due to new challenges within the projects. Indeed, a considerable part of these challenges is related to SCM. In comparison with local single site projects, the distributed projects are clearly hampered by the lack of SCM-related information, concerning such issues as release building schedules, changes in interface modules, approval and rejection of changes, and side-effects of changes. Furthermore, the varying SCM practices and tools are complicating the co-operation between the teams. This will also complicate follow-up tasks as the SCM infrastructure is not mature enough. In conclusion, due to these reasons the company has not been able to make optimal use of the time and knowledge of the team members.

Società Italiana Avionica (SIA) was founded in 1969 with support of the Italian Air Force. Today, SIA is part of the FINMECCANICA (ALENIA) group, which holds 50% of SIA's share capital. Other major owners are MICROTECNICA of Turin and TELEAVIO group. In 1998 the net sales of SIA totalled EUR 11.5 million. From the beginning and throughout the 70s avionics systems has constituted the main market sector of SIA. Currently, SIA operates

in military and civil sectors, such as military aircraft, helicopters, defence, space, military vessels, telecommunication, and railways. SIA employs about 220 persons. Most of the employees work in Turin, Italy, while some employees operate in integration sites dispersed throughout Italy (mainly Rome) and Europe (mainly UK and Germany). Moreover, as SIA is involved in many large co-operation projects in the field of avionics (e.g. EF-2000, Euro Fighter Aircraft), and space systems (e.g. system design for launchers, space platforms, and satellites), the projects are generally coalitions of several organisations. Thus, they are in most cases staffed by international teams within a geographically distributed environment, wherein the bonds between the teams and specialised experts can be temporary. Furthermore, SIA itself also operates as a sub-contractor for other companies.

The characteristics of SIA's main market sectors make the company a very challenging environment for SCM. For instance, the security rules and product quality requirements, concerning e.g. reliability, are extremely stringent. Furthermore, SIA are faced with other new SCM challenges related to, for example, SCM infrastructure and facilities. Incompatible tools in different sites, weak inter-working capability of such tools, data transmission, and security problems are just few examples of the current SCM challenges. Furthermore, SIA is facing new challenges concerning the communication between the teams, and the use of different SCM procedures within a single project.

3.3 Defining the SCM Requirements in a VSC through the Case Study

3.3.1 Background

As mentioned previously, to be able to analyse the SCM process within the distinct context of a VSC environment and to make it meet new challenges, the structures and sub-processes of the SCM have to be outlined and understood first. Primarily, the new SCM requirements have to be identified. The case study is about the activities undertaken to define the SCM requirements in the context of a VSC environment. The requirements analysis has been carried out within the VISCOUNT project, with the author of this thesis being responsible for analysing the case environment, providing the methods for the requirements

analysis task, as well as collecting and evaluating the results. The outcome of the requirements analysis constitutes the basis for an expanded SCM process definition and thus also that of a new approach to analysing and developing the SCM process within a VSC environment.

The first section of this chapter establishes the organisational context of the case study. There the main task is to analyse the case organisations and problems at hand, to understand the primary causes of the new challenges facing SCM. The purpose of this section is to describe the SCM requirements analysis steps taken in the VISCOUNT project. Here, the first research question, “*How can the SCM requirements be defined for a VSC?*” is answered by presenting a case example of SCM requirements analysis, together with our experiences and results. The first task here, however, is to define the most suitable method for the requirements analysis (section 3.3.2) and to establish a method of utilising it within the case environment (sections 3.3.3 and 3.3.4). After the requirements analysis task is performed within the case environment, the results are analysed. Moreover, the identified SCM requirements are mapped to the traditional SCM process definition in order to define any imperfections in it in the VSC context (section 3.4). The initial experiences and results of the requirement analysis in the application organisations of the VISCOUNT project are presented in Papers II and IV.

3.3.2 Use Cases - A Practical Method for SCM Requirements Analysis

The groundwork for the requirements analysis method definition started in November 1997 with an study of the requirements analysis approaches. In addition to the need for a practical method of eliciting SCM requirements from a VSC environment, there are also other requirements for the approach:

- Industrial goals need to be identified for the SCM development in the context of the current strengths and weaknesses of each application organisation. More specifically, it is necessary to work out carefully weighed arguments for the SCM development and development efforts necessary within the application organisation.

- ❑ A practical technique is needed for modelling the SCM target processes, so that the descriptions will support communication both within the application organisations and the VISCOUNT project.
- ❑ SCM requirements need to be linked with the target processes, i.e. industrial SCM processes, in order to ensure the traceability of the requirements.
- ❑ Exact specifications need to be worked out to be used as a basis both for the LIFESPAN™ design and for the SCM process development of the individual organisations.
- ❑ A method is needed for supporting the evaluation process. The evaluation includes both SCM tool and SCM process development.
- ❑ Comparable tests cases are needed for LIFESPAN™ user trials and, e.g., for training new employees in the future.

The aims listed above were used as a starting point when starting the process of searching for, defining, and testing a practical requirements analysis approach suitable for our case study. The results and findings have been collected to a technical document on the VISCOUNT project and analysed together with the application organisations, so as to identify the major advantages and weaknesses that needed to be considered in the process. Neither the selection of the approaches nor the evaluation were carried out with any particular method or technique. However, a collective decision was made concerning the VISCOUNT project to apply the use case modelling technique, thanks to its several advantages from the application organisations' point of view. These advantages are discussed in more detail in papers II and IV. Moreover, one of the main reasons for the selection is that the technique has been in use in all application organisations and therefore they had gained practical experience and knowledge of its use. In addition, the case project members of the application organisations considered this approach the most suitable one for modelling the SCM process. Furthermore, due to the fact that a more extensive literature survey would have required much more effort that could be allocated to the task, no such approach was taken in the decision making process.

According to Ivar Jacobson (1995), use-case modelling is an analysis technique for eliciting, understanding, and defining functional system requirements. Use cases may also be used for validating requirements and designs, as test scenarios with which the operation of a new system can be checked. Various definitions and descriptions of the use-case modelling method, as well as experiences of its use for different purposes have been reviewed in several publications, such as (Jacobson et al. 1992, Jacobson 1995, Sutcliffe et al. 1998, Lee et al. 1998). Use cases are textual descriptions of work practices, including goal, context, pre- and post-conditions, as well as other mainly static and qualitative characteristics of the process being specified. Use cases allow the capturing of informal, semiformal, and formal information. Table 1 presents a use case table from our case study. The requirements column (Req.) contains the requirements identified during use case modelling, referred to by the id-number.

Table 1. Use case table.

Use case ID	The identification code of the use case, e.g. P_STD_UC1	Req.
Use case name	The name of the use case. The name should describe the goal of the use case as a short active verb phrase, e.g. Create project standards	
Summary	A short statement describing the goal of the use case in context if necessary, e.g. This use case specifies how the project SCM standards are created in the project.	
Frequency	When and how often the use case is performed? For example, at the beginning of the project.	
Actors/Roles	The roles of the persons or systems working/acting within the SCM process, e.g. the project manager, project developers, the SCM system, etc.	
Preconditions	The expected state of a process or a system and actors before carrying out the use case. What must be available and what should be done before starting the use case, e.g. current practices analysed	
Description	The steps of the use case divided among actors/roles, from the initial action to the goal delivery, and any measures thereafter.	
Exceptions	Exceptional conditions and actions for the normal use case.	
Success Post-conditions	The state of a process or system and actors after performing the use case, e.g. project standards accepted	
Exceptional Post-conditions	The state of a process or system and actors after exceptional use cases.	

3.3.3 Progress of SCM Requirements analysis

The SCM requirements were analysed in three stages. At the beginning of **the first stage**, the SCM processes within the application organisations were analysed and initial user requirements were defined. The main goal was to identify the strengths and weaknesses of the SCM processes in the context of the characteristics of a VSC, to understand the primary causes of the new challenges for SCM. The SCM current state analyses were performed separately in each application organisation by the VISCOUNT project members. The author of this thesis was responsible for the analysis task at Neles. The information was gathered mainly through semistructured interviews of members of ongoing industrial projects including software developers, project managers, quality assurance staff, and process specialists. Moreover, a series of discussions was performed within the VISCOUNT project team. At Neles, an SCM-focused Bootstrap assessment (Kuvaja et al. 1994) was performed. The experiences from the SCM current state analysis at Neles are presented in Paper I. The results of the first stage have been collected and used as an input to the next stages.

The main issue of **the second stage** was to define the SCM requirements resulting from the distribution of the software development. This work was carried out in April–December 1998. At the beginning of the second stage, it was decided that the requirements analysis task would be linked to the comprehensive SCM process improvement cycle within the application organisations. Moreover, by the decision of the Project Management Board (PMB) in April 1998, VTT, i.e. the author of this thesis, was named as the task manager of the requirements analysis task. Initially, debis had been assigned as the task manager. The primary reason for the change was that VTT had gained a lot of experience with the Goal/Question/Metric (GQM) approach (Solingen van & Berghout 1999), which has been agreed upon to be used as a basic method of measuring and evaluating the success of the SCM process improvement activities and the LIFESPAN^(TM) development. Thus, the author of this thesis has been responsible for the planning progress, supporting the application organisations during the task, as well as for collecting and analysing the VISCOUNT project level results. Furthermore, the author was responsible for GQM activities and SCM use case modelling at Neles.

The focus of **the third stage** was on process modelling and SCM metrics requirements. This work was done in March–July 1999. The results were captured in two phases – the process modelling requirements in March–May 1999 and the SCM metrics requirements in May–July 1999. The work proceeded in a fashion similar to the second stage, but the viewpoints of the second and third stages were different; the GQM goals and the SCM use cases were redefined and adjusted separately at the beginning of both stages. The author of this thesis was responsible of the requirements analysis at Neles, like in the previous stages, and the work was managed by her; the practical work, however, concerning the SCM use case modelling and requirements collection was mainly carried out by other employees of the VTT. After the groundwork, the author has been responsible for collecting the requirements and further analysis.

The use case approach was used for analysing the requirements in the second and the third stage. As presented in more detail in Paper II, GQM goals were used as a basis when describing the SCM use cases of the target state. In practice, this meant that the organisations would describe workflows focusing on the processes that they preferred to improve, to make more effective, and to understand better. Thus, use cases have been used to help analyse the problem domain, so as to gain a better understanding of its requirements. Furthermore, use cases have served as a communication tool between project members and interviewers. The example presented in the following section and shown in Figure 3 describes an industrial example of the requirements analysis process, providing a useful illustration of this approach.

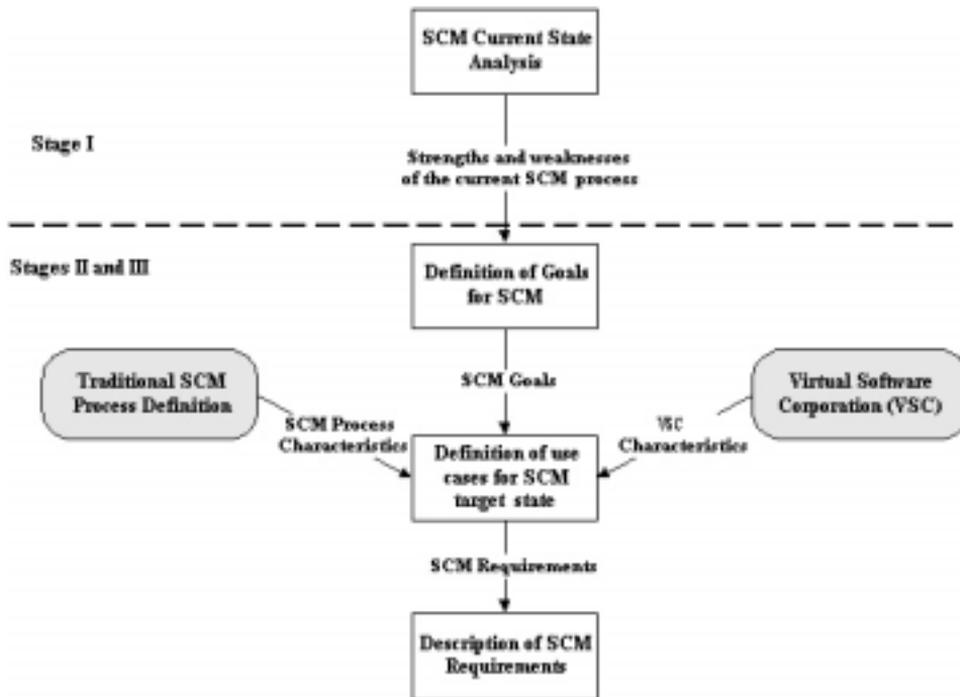


Figure 3. Progress of SCM Requirements Analysis.

3.3.4 An Industrial Example of the SCM Requirements Analysis

The following is a typical example of the SCM requirement analysis work carried out during this case study. It concerns the procedures used for constructing a new release of a product.

During the SCM current state analysis at Neles, it became clear that building a new release of a product in a multi-site project was a very complicated task, causing additional delays and extra work. Consequently, GQM decided to analyse the release manufacturing process in order to improve its effectiveness. To increase the understanding of the process, the process use cases were based on the results of interviewing and brainstorming sessions. The following are examples of defined use cases: 1) Create a failure report, 2) Update the existing SCI, 3) Create a new release for integration test, 4) Create a new release for a customer, and 5) Create a CD for a customer.

During the sessions, some questions have been directed at the virtual project structure, the following being typical examples: “What differences are there between the version building procedures at different sites?”, “Have the sub-contractors been given clear instructions on the version building schedule?”, “Where are the new and altered modules stored?”, and “Do all parties concerned have the necessary rights?”. By analysing the use cases, new requirements for SCM process development have been identified. Table 2 shows an example of a use case table.

Table 2. A Use Case Example: Create a new release for IT.

Use case ID	R_MAF_UC1	Req.
Use case name	Create a new release for IT	
Summary	This use case specifies how a new release of a product is built for integration tests.	
Frequency	Regular release for IT every week and when necessary.	
Actors/Roles	<ol style="list-style-type: none"> 1. Project manager 2. Sub-product managers 3. Chief designer 4. Developers (s) 	
Preconditions	<ol style="list-style-type: none"> 1. Module tests have been performed 2. All the new and altered elements belonging to the release have been checked in an SCM system 3. Binaries have been copied to the release specific build directory 4. Failure reports have been updated 5. Version report for the release has been updated 	
Description	<p>Project Manager:</p> <ol style="list-style-type: none"> 1. Decides on the schedule of the release building process, and the content of each release. 	[R1]

	<p>2. Informs the project members 2 days before release building.</p> <p>Chief Designer</p> <p>3. Checks the changes in interface programs and informs the others if necessary</p> <p>Developers:</p> <p>4. Check that all the elements belonging to the release have been checked in an SCM system</p> <p>5. Copy binaries to the build directory</p> <p>Sub-Product Managers:</p> <p>6. Choose the SCI versions that belong to the release, usually the newest are selected. [R2]</p> <p>7. Check if there are any parallel branches, and combine the elements if necessary</p> <p>8. Check that the compiling and linking options are OK.</p> <p>9. Analyse and check the test run results.</p> <p>10. Move the new sub-product version and other necessary files to the delivery directory and inform other project members of the new release. [R3]</p>	
Exceptions	<p>1. Not all the versions belonging to the release are ready yet</p> <p>2. Not all the versions that belong to the release have been checked in an SCM system. In general, this causes translation errors.</p> <p>3. Other translation or build errors have occurred.</p>	
Success – Post-conditions	<p>1. A new release has been successfully created, and transferred to the delivery directory</p>	[R4]
Exceptions – Post-conditions	<p>1. A new release is built manually.</p>	

As a result, new instructions and a checklist for the version building process have been described, and a programme for automatic and regular night building has begun. The result of this is a reduced number of releases and errors in the process.

The analysis also allows specific requirements to be identified for the SCM system; for example, [R1]: the SCM system has to provide support for sharing derived objects by users between distributed teams. [R2]: The SCM system shall support distributed release manufacturing by applying resources of multiple processors and/or multiple hosts. [R3]: the SCM system shall be able to inform the users when a defined action is performed or activated. [R4]: It shall be possible to easily specify what has been changed in a release. The requirement ID codes (e.g. R1, R2, etc.) have been entered into the use case tables.

3.4 Evaluating the Results

After the requirements analysis tasks, the lessons learned from the main results were analysed. One of the major findings of the analysis concerned the use case requirements analysis approach. Although all application organisations have found the approach rather a laborious way of capturing requirements, they are, however, interested in having a useful technique for obtaining useful information of the SCM process. The experiences gained with the use case approach are presented in more detail in Papers II and IV.

The main goal of the requirements analysis task from the viewpoint of this study has been to analyse the SCM process in the context of a VSC environment, so as to define more accurate new requirements, and thus to arrive at a more expanded content of the SCM process. In all, 92 different use cases and a total of 122 SCM requirements have been defined in the VISCOUNT project. The organisation-level requirements are quite detailed, and they have been unified and harmonised for the project level requirement list. Hence, the actual number of requirements is greater than 122. The use of various SCM tools by the application organisations in their case projects has led to the fact that the requirements overlap and that they are partly at different levels. However, the aim has been to identify all the requirements of the application organisations, so as to get as comprehensive a picture of the situation as possible. Therefore, the requirements

list may well include SCM tool features that can already be included in a commercial tool. In that case, the specific tool has not been used by the organisation in question, or a specific feature of the tool has not been fully utilised. On the other hand, these requirements have not necessarily been implemented in LIFESPAN™, and are thus important sources for the tool development. A summary of the key findings regarding the identified SCM requirements in the context of the SCM base processes already identified in Chapter 2 Section 2.3 (i.e. configuration status accounting, configuration identification, change control, configuration auditing, release manufacturing and management) are presented below.

Configuration status accounting. The analysis of the SCM requirements clearly indicates that the virtual environment considerably increases the need to get more reliable and real time information on the SCM process. In other words, as a part of a more extensive effort to try to manage temporary teams working in dynamic and distributed environment, the attention in the application organisations seems to be focusing on collecting, analysing and reporting SCM-related data. In our study, 17 specific requirements directly related to the need to make SCM more visible have been identified. Most of the identified requirements are also typical in the traditional single site organisation; such are, for instance, reports relating to documentation (size, changes and the size of changes) and source code (number of source files, number of revisions to source files). However, when working within a VSC, the definition, collection, and analysis of SCM data proved to be much more complicated than in a single site environment. The challenges of SCM data collection and analysis are discussed in more detail in Paper VI.

Configuration identification. The analysis of the requirements concerning configuration identification indicates that the current SCM tools do provide adequate support for the application organisations in most cases. However, three specific requirements were identified for configuration identification; these are related to distributed development and multi-platform environment (version identifier and site information, multi-platform support, and management of dependencies over distributed sites).

Change Control. The change control process takes care of all changes concerning SCIs throughout the product lifecycle. In all, 19 specific

requirements concerning the change control process were identified. The analysis of the requirements reveals that although the change control process is quite well established and repetitive in the application organisations, and the process is supported by advanced SCM tools, there are still some significant improvements to make. The findings show that the main difference between traditional systems and VSCs is flexibility, which is likely to lead to greater complexity. To be able to utilise the flexibility of the virtual structure effectively, appropriate methods and tools have to be found for implementing the resulting changes in a fast and effortless manner. Otherwise, the benefits gained would be lost in an increased organisational effort necessary for steadily updating the mechanisms used for controlling the increased complexity. This is demonstrated by the number of requirements that have to do with tailoring and automating the change control activities. This again implies that it takes a good understanding of the virtual SCM process as a whole to be able to formalise the processes for automatic enactment. Furthermore, as also stated in (Conradi & Westfechtel 1999), there is a need to further examine the dependencies and interfaces between SCM and the specific process tools.

Configuration auditing. Most of the requirements related to change control process, and six of the requirements concerning configuration auditing, are focused on process automation and tailorability. In both cases, it is evident that the VSC-specific characteristics – distribution, impermanence, dynamism, and more complicated teamwork activities – lend diversity to the processes, and therefore have to be tailored to fit the needs of the projects. However, as it has been pointed out by the case project members of Neles, the projects rarely have the time or the competence to tailor the SCM process themselves. Rather on the contrary, tailoring should belong as an integral part to a general SCM process improvement program, in which the SCM measures make it easier to evaluate the results of the actions.

Release manufacturing and management. Although the advanced SCM tools of today provide quite advanced support for parallel development, allowing, e.g., branching and merging operations, the release manufacturing and management process still seems to be highly dependent on manual processes. The process typically comprises a number of different phases, which will become even more complicated within a distributed, dynamic environment. In all, 12 specific requirements related to the release manufacturing and management processes

were identified in our study. Most of the requirements focus on SCM tool features, such as "an SCM tool shall explicitly retain release interdependence", and "an SCM tool shall support derived object sharing between users". However, as has been indicated by the Neles case (see e.g. Papers I, II, III), the SCM challenges focusing on release manufacturing and management have been largely brought about by the lack of process and communication support.

The analysis of the identified SCM requirements clearly demonstrates that the SCM base processes within the application organisations mainly lack the infrastructure for tailoring and managing the processes. Moreover, the integration of these activities to a comprehensive process improvement program has been emphasised in many discussions with the application organisations. This conclusion suggests that an **SCM improvement process** tightly connected with the base processes could offer an appropriate solution. This is an aspect of great importance within VSCs, as in these not only the process but also the organisation, methods, and tools are under continuous change. The content and background of the SCM improvement process will be elaborated upon in the next chapters.

The analysis of the identified SCM requirements has revealed that more than half of the requirements (65) are focused on SCM-related planning, management, and teamwork activities rather than existing base processes. The initial current state analyses in the application organisations also support this view. Moreover, many of the listed requirements have been identified by more than one of the three application organisations. Although the distribution, impermanence, teams, and dynamism characteristics of the VSC have a distinct effect on the SCM base processes, the new major SCM challenges are clearly concerned with a more extensive context. Thus, one of the key lessons learnt from the SCM requirements analysis is that an expanded SCM process definition is needed. The traditional definition seems to be inadequate for the SCM process analysis and development within a virtual environment. Thus, additional SCM processes are required. Another key lesson is the recognition of the importance of the SCM-related planning and management activities and thus the realisation of the growing need for making the SCM process more visible and more predictable. This expanded SCM process, as well as the ideas and the key requirements behind it are presented in more detail in the next chapter.

4. Towards Virtual SCM Process

4.1 Introduction

This thesis is concerned with SCM, software configuration management, within the specific context of VSCs, i.e. In Chapter 2, the application organisations of our study are examined with reference to the characteristics typical of VSCs and SCM. The study reveals that the increasing complexity of virtual organisations presents new challenges to the SCM process. However, in order to properly analyse the SCM process and to improve it, so as to make it meet these new challenges, the activities of the SCM processes found in a VSC have to be recognised first. In other words, an expanded SCM process must be defined to serve as a starting point. This requires, however, accurate SCM requirements to be identified and evaluated against the existing assumptions of the SCM process.

Chapter 3 presents the requirements analysis approach applied in our study and an overview of the main results. The generic applicability of the research results is guaranteed by the fact that the study is not based on a single case but on the case studies of three application organisations differing from each other, for instance, in terms of company size, net sales, and portfolio. Thanks to the wide scope of the SCM requirements, the results of the SCM requirements analysis form a good basis for an expanded SCM process development presented in this chapter, which addresses the second research question “*How can the SCM process be expanded for a VSC?*”.

4.2 Lessons Related to the SCM Challenges

4.2.1 Background

The core activity of a VSC is the production of software. The subject, structure, and technical aspects of the software product determine the composition of the VSC team, i.e. independent institutions, enterprises, and specialists are selected for a specific purpose. From the viewpoint of SCM, the product domain can be viewed as the *SCM environment*. In our study, SCM environments vary depending on the portfolio of the application organisation. For instance, the space and avionics products of SIA represent highly complex systems developed

in large, international multiorganisations, involving a wide application area, extensive technical knowledge, and a need for highly qualified experts. Contrary to this, some small maintenance projects of Neles, which also show the typical characteristics of virtuality, represent the other end of the spectrum.

An established *Virtual Software Corporation (VSC)* runs the *SCM process* by means of *SCM techniques*, certain methods and tools that must support the specific needs of the SCM within a particular VSC. *Figure 4* illustrates the position of the SCM process in the context of SCM environment, VSC organisation, and SCM methods and tools. From this viewpoint, the experiences concerning the SCM process and the used SCM techniques in the application organisations have been collected and analysed by the author of this thesis. The findings clearly indicate that virtuality presents new challenges to the SCM base processes as described in the previous chapter. Basically, this alone means that the SCM processes used in the application organisations needed to be developed further along with the SCM tools and methods. However, the main lesson learnt is that most of the identified SCM requirements do not focus on the traditional base processes. This fact, along with the literature survey and the results of the SCM current state analysis of the application organisations, provides a good starting-point for the development of an expanded SCM process description.

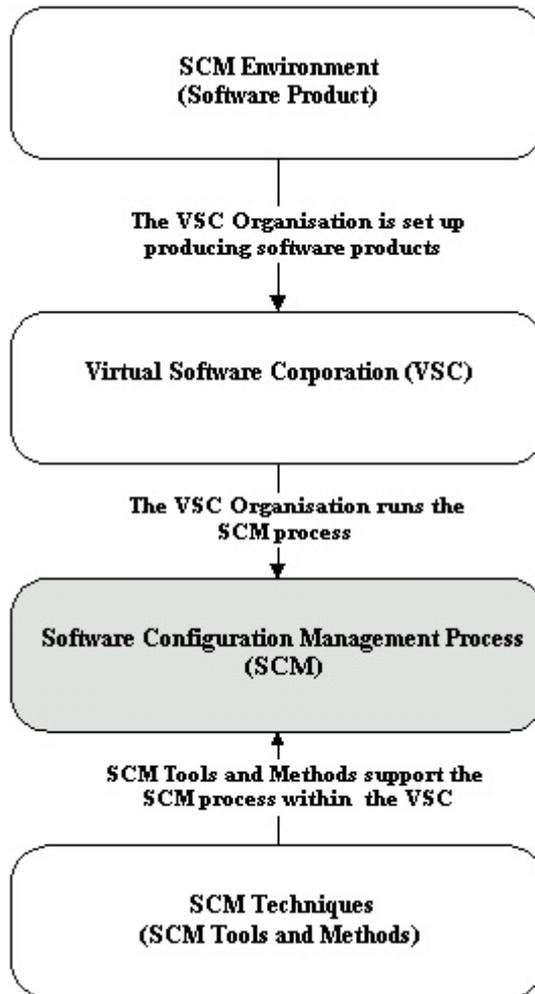


Figure 4. The position of the SCM process.

The groundwork for the expanded description of the SCM process started during the SCM current state analysis phase in early 1998, when the author of this thesis started to look for the root causes of the SCM challenges in the application organisations. An analysis of the related work (Chapter 2) was used to extend the basis. However, the main work for the expanded SCM process development is based on the studies carried out during the SCM requirements analysis task. The following sections demonstrate how the virtual environment affects the SCM process, and presents the lessons learnt from our study concerning the new SCM challenges. For more detailed information on the identified SCM requirements see the VISCOUNT project web page, <http://www.viscount.org>.

4.2.2 SCM Improvement

The existence of different software products, the composition of the VSC, along with the multitude of projects types, SCM tools and methods makes it impossible to apply any uniform and stable SCM processes across all VSCs. Quite the opposite, it seems that the tailoring of SCM procedures, methods, and tools individually for each specific purpose will serve virtual organisations better. In our study, for instance, several requirements focusing on the flexibility of SCM procedures and tools have been identified, such as those concerning the generation of templates and forms, the customisability of the life cycle model of the project, state transitions, and data distribution models. It is to be noted here that the prerequisites for tailorability are a sound understanding of the VSC environment and the establishment of an overall framework. Moreover, requirements concerning the continuity and persistence of SCM arose from the discussions with the application organisations. debis, especially, is particularly keen on utilising SCM knowledge in their projects in large. To summarise, the findings from the SCM requirements analysis suggest that an appropriate solution would involve expanding the traditional SCM process definition into an SCM improvement process, which would be tightly interconnected with the SCM base processes. While the notion of continuous SCM process improvement would provide a valuable aspect also within the traditional single site environment, it is especially important within the VSC, due to its impermanent and dynamic nature. In Chapter 5, the SCM improvement process is discussed in more detail, and a practical method is introduced for improving the SCM process for VSCs.

4.2.3 SCM Teamwork

SEI, the Software Engineering Institute used the concept of SCM teamwork to define the features related to the technical viewpoint of the co-ordination of software engineering teams (CMU/SEI-92-TR-8 1992). Here "teamwork" is used to denote specific activities that are required for making a VSC operate more reliably and efficiently. Thus, the SCM teamwork process definition presented here expands earlier conceptions of SCM.

In section 2.4, the VSC related SCM challenges are categorised as follows: infrastructure, facility, security, culture, communication, and switching. These

categories are based, on one hand, on the findings and evaluations during the SCM current state analysis of the application organisations and, on the other hand, on the results of the literature survey. The experiences with the identified SCM requirements consolidate and expand our concept of the *SCM teamwork* sub-processes:

SCM Infrastructure. SCM Infrastructure provides the basis for effective teamwork. However, the virtual structure of an organisation makes it very difficult to plan, to implement and to maintain the infrastructure. Particular characteristics of a VSC, such as distribution and impermanence, tend to cause such changes in the SCM infrastructure that are not often faced by single software development organisations. The findings of our study suggest that traditional SCM process descriptions lack the aspect of process support for infrastructure. Moreover, the identified SCM requirements clearly indicate that the distribution and impermanence characteristics cause new SCM challenges that can not be solved using existing procedures, practices, and tools. For instance, the identified SCM requirements, e.g. concerning network architecture, data transmission across distributed teams, and interfaces to other tools (e.g. other SCM tools, fault databases, customer call tracking systems, project management tools, and quality management tools), are issues that cause problems to current industrial VSCs. Thus, on the basis of our study, it is suggested that when analysing and improving the SCM within the distinct context of a VSC environment, attention should be paid to the SCM infrastructure, i.e. network architecture, host development, and target-built platforms and tools (development tool sets, support tool sets, operating systems).

SCM Facility. The increasing importance of proper technical support for distributed teams can be clearly recognised in the related literature. However, as stated earlier (in Asklund et al. 1999, Estublier 1999, Frühauf & Zeller 1999), no single currently available SCM system is capable of providing solutions to all SCM problems related to distribution and impermanence. On the contrary, the findings from the SCM requirements analysis suggests that the organisations are lacking in proper planning of 'SCM Facility' related activities. For instance, although the SCM tools would provide functionality concerning parallel development, concurrent changes, and merging, they may not be fully usable (e.g. results from SCM current state analysis at Neles, Paper I) or are not suitable in all cases (e.g. when a team does not have a permanent network connection, as

identified during the SCM requirements analysis tasks). This finding does not necessary mean that all SCM tool features would have to be fully used, but rather that the corresponding activities should be planned properly. The lessons learnt indicate that it should be carefully planned what SCM tool functions are used and how they are used, what other facilities are used, what attributes the systems will have, and what the performance requirements for the SCM facility are. For example, it should be possible for the organisations to use several different techniques for distribution. Thus, as a part of this revision of the SCM process description, the SCM facility sub-process is defined. 'SCM Facility' involves establishing requirements for SCM techniques (functions to perform, attributes the system will have, performance requirements), acquiring resources (hardware, software, space, and people), and implementing new features needed.

SCM Security. The lessons learnt concerning security issues confirm the fact that for VSCs, security is an issue of great importance. Security calls for constant attention due to its criticality in many instances. First, there is the issue of securely distributing large amounts of data in a timely fashion over great distances (Buffenbarger & Gruell 1999). Furthermore, the identified SCM requirements indicate that all the application organisations have concerns about the security of data transmission across public networks. However, the security issue is not only a technical question, in which the concerns of the organisations focus on distribution techniques, data compression and encryption, as well as signatures (e.g. Noll & Scacchi 1997, Buffenbarger & Gruell 1999). SCM security is a critical issue also at the organisation level. The survey reveals that the processes of planning and implementing security rules and guidelines needs to be enhanced. Some of the essential questions to tackle would be, for instance, what information could be disseminated to the sub-contractors and other partners, who should be allowed to make changes to the documentation, who should be allowed to make changes to specific modules, who should define the security levels, and how the security issues should be implemented. These questions are explicitly raised during the survey. Therefore, it can be suggested that the scope and definition of the SCM process should be expanded to include 'SCM Security' as a sub-process, i.e. defining security requirements, restrictions, and related practices and procedures within the VSC, and implementing the solutions needed.

SCM Culture. One of the challenges identified in the SCM current state analysis and literature survey has to do with the culture-specific issues connected with SCM, which are closely related to the teams and dynamism characteristics of a VSC. Single software development teams do not normally have to face any culture-specific differences, including different SCM methods, tools, and techniques (Noll & Scacchi 1997, Van De Vanter & Murer 1999). Upon closer scrutiny of the matter during the SCM requirements analysis task it became evident that in many cases the culture-specific issues have caused problems within our application organisations as well. For instance, when a sub-contractor is using quality procedures different from those of the main company, misunderstandings may cause extra work for all. On the other hand, it is not always possible for all the teams to use similar procedures and tools. An indication of this could be seen in the requirements concerning the interfaces between different tools. One of the main lessons learnt here is that various compositions of procedures and tools may be in use within a single VSC. However, to guarantee an adequate level of efficiency and reliability, a VSC should be able to combine these procedures and tools in an optimal way. First of all, it is necessary to know all the procedures and tools used by the parties involved, as well as their interfaces. Secondly, the rules and guidelines for their use in the VSC should be planned carefully, as the findings from our study show that undocumented practices are a risk to the process, since they are hard to maintain when the resources available are changing. Thus, the 'SCM Culture' sub-process is proposed, focusing on recognising the culture that exists within the VSC and defining culture-specific requirements for the practices and procedures of the SCM process.

SCM Communication. One of the key issues revealed by the survey is the need for developing and improving the communication between the teams. The communication aspect can be explored from various viewpoints and there are also a number of other essential related issues to be considered. This study concentrates on SCM specific communication. While the teams, in most cases, are large and the people involved are distributed across different sites, the communication is, no doubt, much more complicated than in a single site environment. The findings of the SCM requirements analysis task indicate that there is a clear need to improve the communication mechanisms within the VSCs. For instance, additional ways of communication should be provided, such as electronic news servers under an SCM tool, along with external e-mail

integration. Moreover, to make communication more efficient, the planning of communication rules and procedures should be enhanced. One practical example is provided by the changes made to the interface programs. When a change is made, the people implementing the related programs should always be informed. The identified SCM requirements also revealed the importance of automation. The need for better tool support concerning SCM communication is thus an important issue also for the tool developers. In conclusion, the identification, implementation, and management of communication is an important task, concerning both the SCM process and technical issues. Thus, the 'SCM Communication' sub-process is defined, focusing on planning and implementing communication practices and procedures between the distributed teams within a VSC.

SCM Switching. An effective utilisation of resources, and performance optimisation are the basic management principles of a VSC (Mowshowitz 1997). One of the main lessons learnt from the study is that these issues are closely related to SCM. In all, the SCM process produces and includes information on what tasks are to be performed, the states of the tasks, as well as on resources, roles, and responsibilities. This information forms the basis for the switching of resources (human, process, or technical resources). In other words, SCM creates a framework for the activities, which makes it possible to share tasks dynamically, to borrow know-how as needed, to use experts efficiently, and to balance resource load. Although the VSC types and resource coalitions in the application organisations of our study are different, the importance of resource switching has been identified in all cases. The findings of the SCM requirements analysis also support this view. It is required, for instance, that task switching should be planned properly and that it should be supported by a systematic use of methods and tools. Moreover, some of the comments indicate the importance of task management issues, such as the need of receiving information on forthcoming tasks, utilisation of resources, interests areas of resources, special skills of staff, and the like. The traditional SCM seems too much focused on rough data, while it is lacking the aspect of how to make use of SCM related information, which is essential since most of the resources operate in a rapidly changing, dynamic teamwork environment. Therefore, the 'SCM Switching' sub-process is proposed for defining the practices and procedures for switching and managing tasks, and using them to improve the utilisation of resources within a VSC.

SCM Management. The teamwork related SCM sub-processes focus on identifying the activities that a VSC needs to cover in order to operate reliably and efficiently within the distinct context of a VSC environment. The original idea of these sub-processes includes six categories related to the SCM infrastructure, facility, security, culture, communication, and switching of resources. However, a further discussion which took place towards the middle of 1999 in a Neles feedback session, indicated that this definition was insufficient. The Neles project members pointed out that in too many cases valuable process improvements had failed or got drawn out due to the lack of proper planning and management. Moreover, the findings from the SCM requirements analysis also demonstrate the importance of the management aspect. For example, the planning and follow-up of the SCM activities are considered significant. Thus, due to the concern about the established SCM sub-processes not being properly utilised due to the lack of planning, support, and management, the SCM teamwork process is supplemented with the 'SCM Management' sub-process. This sub-process has to do with managing human, technical, and process resources within the VSC, guided by the results of the SCM planning process, which is discussed in the next section.

4.2.4 SCM Planning

The purpose of SCM planning is to define, to monitor, and to control SCM practices, as well as to ensure that software is developed according to a defined software development methodology (ISO/IEC-12207 1995). As a result of the SCM planning process an SCM plan for a VSC is worked out. An example of the content of a traditional SCM plan is described in the IEEE standards (IEEE-1042 1987).

In addition to the SCM base processes, the SCM planning process must be expanded, for a virtual environment, to cover teamwork activities as well. Generally, the SCM plan includes practical instructions and guidelines on how to carry out SCM activities. The activities within a VSC include SCM teamwork, configuration identification, configuration status accounting, configuration auditing, release manufacturing and management tasks, as well as accomplishing these tasks after the VSC has been dissolved and the product is in the maintenance phase. For instance, it is determined how many configuration audits are to be performed in each site of the project, what is audited, and what

material is to be reviewed. Moreover, the various organisational and user roles and their responsibilities related to SCM are settled upon. These roles involve various functional roles such as developers, testers, documenters, QA people, project managers, technical support, and SCM tool support within the VSC.

In addition to the activities, roles, and responsibilities, the standards and techniques for the VSC are planned, and implemented if needed. The standards include, for instance, the naming, coding, and version reporting standards that are to be used in the VSC project. Furthermore, it is decided which SCM tools and methods are to be used within the VSC, and which SCIs will be subjected to them and where they will reside.

The information on SCM planning provided in this dissertation is given in outline form only. The reason for the brevity is that more extensive definitions and guidelines for SCM planning are presented in several publications, such as (Buckley 1996, IEEE-1042 1987). Even though SCM planning within a virtual environment is more complicated than in the traditional single site environment these publications can be utilised within a VSC as well. However, in this case, special attention should be paid to the SCM teamwork aspect. In other words, the activities of the teamwork process should be included in the planning process as well. Furthermore, the SCM base processes should be planned in consideration of the teamwork aspects and the virtual environment. For instance, when planning the procedures for release manufacturing, the SCM infrastructure and facility issues within different sites of the VSC should be considered. Furthermore, when planning the configuration auditing sessions, the issues having to do with SCM communication are crucial. The relationships between the SCM sub-processes are discussed in more detail in the following section.

4.3 The Expanded SCM Process Description

Figure 5 below represents a new, expanded definition of the SCM process, constituting SCM within a VSC as discussed above. One of the greatest challenges for SCM in a VSC is to manage the teamwork activities, especially communication and task co-ordination within distributed teams. Therefore, *SCM Teamwork* is defined as a special process affecting the *SCM base processes* shown in Figure 5. The other processes are *SCM Planning*, by which the SCM

teamwork and base processes are adjusted to the needs of specific software development projects, and *SCM Improvement* by which the SCM is continuously analysed, measured, and improved within a VSC.

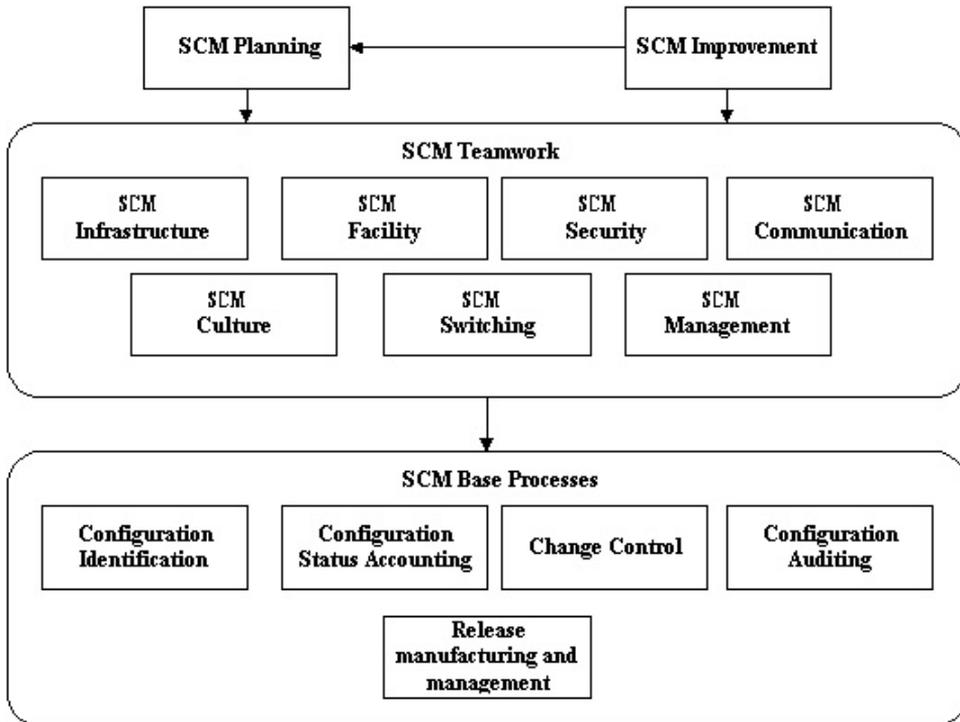


Figure 5. SCM Teamwork, SCM Improvement, SCM Planning, and SCM Base Processes.

In the Figure 5, the arrows from the SCM planning and SCM Improvement boxes go down only to the SCM Teamwork rectangle. This does not, however, mean that the SCM planning and improvement would not be directed towards the SCM base processes. On the contrary, the purpose is to highlight the fact that when planning and improving SCM activities within a virtual environment, the base processes should be considered through the teamwork processes. Accordingly, the arrow between the rectangles of SCM Teamwork and SCM base processes, illustrates this view. Moreover, the arrow puts emphasis on the fact, discussed in Chapter 3 Section 3.4, that the SCM base processes within a virtual environment differ from those in a traditional single site environment. This difference is also demonstrated by the arrow between SCM Teamwork and

SCM base processes. The SCM Improvement process is targeted on all the other SCM processes. The activities of the SCM Improvement process are discussed in more detail in Chapter 5.

4.4 Conclusions

The major challenge for developing SCM to meet the requirements of VSCs is to recognise what the specific elements involved in VSCs are. By analysing the current status of SCM in the application organisations, it has been realised that this challenge can only be met by increasing our understanding of the virtual SCM process. It has also become evident that most of the SCM challenges are related to the SCM Teamwork process and its sub-processes (infrastructure, facility, security, culture, communication, and switching), rather than traditional SCM base processes. As a starting point of our study, the current strengths and weaknesses of the SCM processes within our application organisation were analysed and a survey was carried out on literature concerned with issues connected with SCM and VSC. The use case approach was employed in analysing and defining SCM requirements.

The results of the SCM requirements analysis revealed that there is not much benefit from the existing SCM process descriptions, as most of the identified challenges and requirements concerning SCM are related to SCM management, teamwork, and improvement activities, rather than the traditional SCM base processes. Therefore, the SCM process had to be expanded and elaborated. The expanded definition for the SCM process includes SCM base processes and an expanded SCM planning process, as well as processes concerning SCM teamwork and improvement.

5. Developing SCM to meet VSC Requirements

5.1 Introduction

Many large software projects now underway will face significant challenges concerning the software configuration management (SCM) due to the changes in the software production environment. Software will be produced increasingly within virtual software corporations, in which everything may be distributed (Boldyreff et al. 1996, Noll & Scacchi 1997). The SCM current state analysis of our application organisations, as well as the literature survey and the a SCM requirements analysis, suggest that one of the largest single cause of the challenges is a poor understanding of the new demands made on SCM. In our study, the traditional SCM process description has been expanded to cover new areas of SCM, so as to enable SCM process analysis within the distinct context of a VSC environment. The purpose has been to provide a basis for SCM process improvement. In this chapter, the third research question “*How can the SCM process be developed to meet the VSC requirements*”, is answered by presenting an approach for SCM development in view of the VSC requirements. In the approach, we combine and utilise the research results presented in the previous chapters. Furthermore, we depend on practical experiences collected during our study, which are presented in the accompanying original papers.

Software process improvement (SPI) is seen as an important solution to meeting the challenges in software development, and a large amount of research and practical work has been done in this area (Brown 1996, Fitzgerald & O’Kane 1999, Kinnula 1999). The interest in process improvement is also indicated by the success of SPI methods and tools such as Bootstrap (Kuvaja et al. 1994), CMM (Paulk et al. 1995), and SPICE (Eman et al. 1998). Furthermore, new SPI approaches are being developed, such as the Profes (Product Focused Software Process Improvement) methodology (Birk et al. 1998), which concentrates on product quality in software process improvement. An example of SCM-focused process improvement is presented in Taramaa’s dissertation “Practical Development of Software Configuration Management for Embedded System” (Taramaa 1998). A similar approach has been also introduced in (Kilpi 1998). The results presented in these research studies are based on experiences from

either small or medium enterprises (SMEs), or small software engineering groups within larger companies. However, our experience indicates that VSCs require useful and practical strategies, methods and techniques that focus on SCM process analysis and development, especially in the context of the virtual process. This is stated also by Taramaa when presenting his ideas on further SCM research topics (Taramaa 1998).

5.2 Improvement Framework

The improvement framework applied and further developed in our study is called Pr²imer, *Practical Process Improvement for Embedded Real-Time Software* (Karjalainen et al. 1996, Komi-Sirviö et al. 1998). Pr²imer is an improvement action cycle model, which gives guidance on how to proceed in executing a software process improvement program. The framework divides improvement activities into four main phases, each with a separate focus. The main results of the phases are defined as well as the flow of information from one phase to another.

In Chapter 4 Section 4.2, the idea of the 'SCM Improvement' process is introduced, since, it has been considered significant in our study that the improvement aspect would be tightly interconnected with SCM. In this work, the focus of the Pr²imer approach is on SCM within the VSC environment. Therefore, our approach is called *SCM-Pr²imer* for VSCs and it is presented as the 'SCM Improvement' process, which has been identified previously. SCM-Pr²imer addresses the SCM-related activities of the software process by supporting the four phases presented in Figure 6. This is quite similar to the "standard" Pr²imer improvement cycle, except for the phases being concerned solely with the SCM-specific aspects of the software process within VSCs.

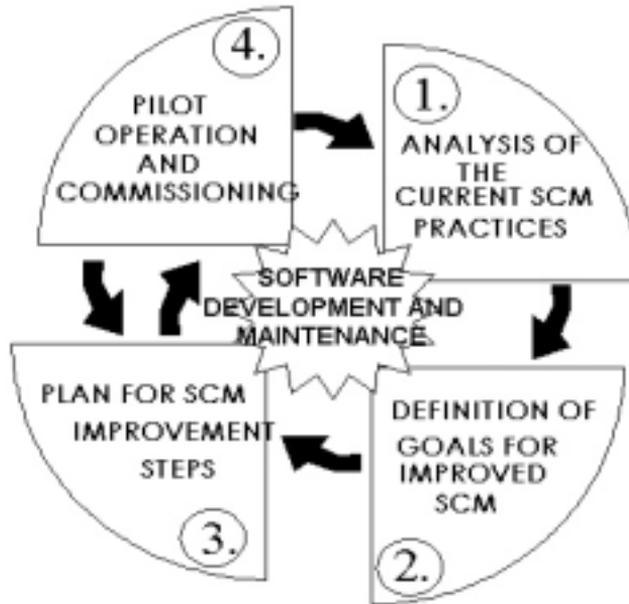


Figure 6. *SCM-Pr²imer (Paper I)*.

The purpose of this chapter is to present an SCM-focused approach intended for guiding an improvement project for developing SCM in order to meet the VSC requirements. The SCM-Pr²imer framework for VSCs is similar to the basic structure of the Pr²imer. The differences can be found in the *first* and *second* phases. The first SCM-Pr²imer phase is the most crucial one. It comprises the current state analysis phase and it forms the basis for the following phases. Here, the results of our study are utilised when employing the expanded SCM description as the basis for analysis. The purpose of the second phase of SCM-Pr²imer is to identify goals for the SCM process improvement and to describe a target state for the SCM process within a virtual environment. In the second stage, we apply the use case approach presented in Chapter 3. All the SCM-Pr²imer phases are introduced as a short summary in the following sections. The practical experiences and more detailed descriptions of each phase are presented in the accompanying papers as follows: *SCM-Pr²imer phase 1*: Paper I, *SCM-Pr²imer phase 2*: Paper II, IV, *SCM-Pr²imer phase 3*: Paper III, VI, *SCM-Pr²imer phase 4*: Paper V.

5.3 SCM Current State Analysis

5.3.1 General

In general, SCM-Pr²imer employs various methods and techniques, the selection of which is based on the specific situation and the needs of the particular organisation. The selective nature of SCM-Pr²imer can be seen especially in the first phase, in which current practices are analysed and evaluated by means of quantitative and qualitative analysis methods.

A quantitative analysis can be carried out using the SCM-specific parts of the Bootstrap (Kuvaja et al. 1994), CMM (Paulk et al. 1995), and SPICE (Eman et al. 1998) assessment methods, for example. The SCM maturity levels described in (Taramaa 1998) and (Kilpi 1998) can also be employed in forming an approach, which can be used for focusing on the SCM-related elements of the process. However, all the above methods are based on traditional SCM process definitions, and as the work reported here in Chapters 3 and 4 has shown, they fail to cover the new challenges for the SCM process in VSCs, such as those related to the SCM infrastructure, facility, security, culture, communication, and switching of resources.

The qualitative analysis is based mainly on discussions with the members of a particular case project or projects. In addition, also other persons participating in product development, such as managers, quality engineers, and experts, should be interviewed. The general aim is to analyse the methods and tools that are currently used in the organisation, and to identify the strengths and weaknesses in the process. Furthermore, the current supporting techniques, such as guidelines and templates, are evaluated.

The practical experiences from a current state analysis in an application organisation are presented in Paper I. The analysis is based on a traditional SCM process description, even though the idea of a more comprehensive SCM analysis is introduced. During our study, it was, however, realised that an expanded SCM process description was needed. A current state analysis within a VSC differs remarkably from one made in a traditional single site environment. The processes to be analysed are more complex, consisting of a web of firms, teams, individuals, and technologies. In fact, the analysis cannot be carried out

properly without a thorough understanding of the new challenges for the SCM process. Therefore, the procedures for an executive analysis of the current state of SCM in a VSC environment have been expanded and described more accurately.

5.3.2 Applying an Expanded SCM Description in SCM-Pr²imer

This section gives an introduction to using the expanded SCM process description in the current state analysis of SCM-Pr²imer. To be able to employ the SCM-Pr²imer framework, the business and management issues that give meaning and rationale to the program must be identified first. In other words, understanding the business strategy and goals is a prerequisite for all improvement actions. From the viewpoint of SCM development towards a virtual SCM process, the most important aspect here is to identify and to understand the SCM environment and the VSC structure (see *Figure 4* in section 4.2). The systematic procedure of the SCM-Pr²imer program, from the very beginning on, is presented in *Figure 7* and described briefly below. *Figure 7* illustrates the steps used for VCSs, which are similar to those of the standard Pr²imer method, as described in (Karjalainen et al. 1996 and Komi-Sirviö et al. 1998), except for the activities concentrating on SCM. Moreover, the described steps originate from the GQM approach (Solingen van & Berghout 1999), which is one of the principal methods the Pr²imer improvement cycle is based upon.

The three main steps of the first SCM-Pr²imer phase are: 1) identify and understand the SCM environment, 2) understand the business strategy and goals of the VSC, and 3) analyse the current status of the SCM process. These steps are described briefly below:

Identify and understand the SCM environment. The first step in the SCM current state analysis is to identify and to understand the SCM environment in question. In other words, the software product under development and its basic elements must be determined first. For example, in one of the application organisations of our case study, a planning session was held at the beginning of the improvement program, the aim of which was to collect background information from the company personnel, to be used by the improvement team, on such issues as technical environment, purpose of use, customer profile, and the market.

Understand the business strategy and goals of the VSC. A VSC is a teamwork environment, where all the members share a common goal. Therefore, to be able to successfully initiate an SCM improvement program, the underlying business strategy and goals of the specific VCS must be first clearly identified, agreed upon, and fully understood by the entire VSC.

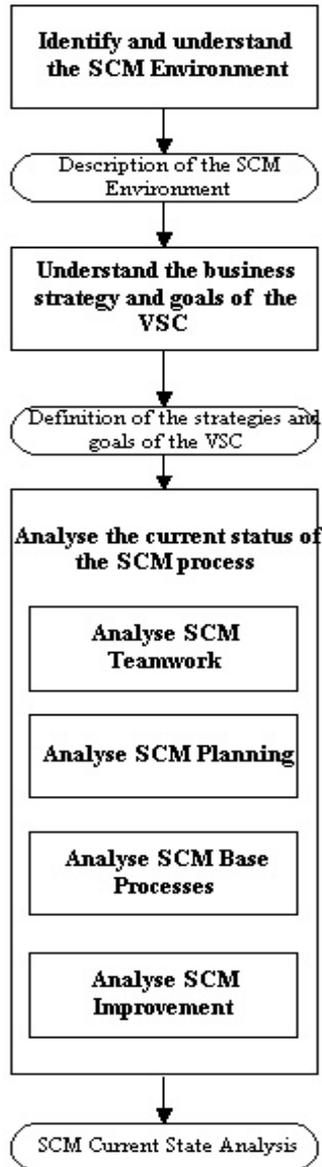


Figure 7. The procedure of the first SCM-Primer phase.

Analyse the current status of the SCM process. An exacting task in developing the SCM towards a virtual structure is to understand and to define the current status of the SCM process. Several analysis methods may be applied in the SCM analysis, depending on the needs of the organisation. Most of the methods are based on discussions and interviews with the case project members. Generally, based on our experience, the information derived from the people working on ongoing projects is likely to give the most reliable picture of the overall situation. Anyway, to ensure efficiency, it can be recommended for the interviews to be based on a systematic view of the SCM process. Therefore, the expanded SCM process description developed during our study was employed to support the analysis task. In other words, the process description offers a framework for the interviews, while also ensuring that all the relevant aspects will be taken into account. Table presents the structure used when analysing the current status of the SCM Teamwork sub-process. The questions under each heading are specifically used to provide the qualitative and quantitative information upon which the analysis can be based. The questions presented in the table are derived from the SCM requirements identified in our study. Only the SCM Teamwork table has been presented here to illustrate the approach. Deriving similar checklists for other SCM sub-processes is the next step for the future research.

Table 3. The SCM Teamwork process and a question list.

SCM Teamwork	
Culture	<ul style="list-style-type: none"> <input type="checkbox"/> Has the VSC environment been described? <input type="checkbox"/> Are there any culture-specific restrictions within the VSC? <input type="checkbox"/> How many languages are used? Has the VSC level 'official' language been defined? <input type="checkbox"/> What kinds of SCM standards are in use in the different teams? <input type="checkbox"/> Time zone differences? <input type="checkbox"/> How many teams? Team structure?
Infrastructure	<ul style="list-style-type: none"> <input type="checkbox"/> Has the network architecture been described? <input type="checkbox"/> What kinds of procedures have been defined for a new team? <input type="checkbox"/> What kinds of platforms are in use? <input type="checkbox"/> Tools (development tools, supporting tools, SCM tools)
Facility	<ul style="list-style-type: none"> <input type="checkbox"/> What data distribution techniques are used in the SCM system? Does the SCM system enable selective replication? <input type="checkbox"/> Does the system support automatic distribution? <input type="checkbox"/> How does the SCM system support data transmission? Additional ways? <input type="checkbox"/> How many different SCM tools are in use? Interfaces between the tools? <input type="checkbox"/> SCM system integration with the other tools? <input type="checkbox"/> Does the SCM system support parallel development, concurrent changes and merging? <input type="checkbox"/> Performance requirements of the SCM system (hardware, software, space, people)?
Switching	<ul style="list-style-type: none"> <input type="checkbox"/> Is it possible to reassign the responsibility for tasks between project members and teams? <input type="checkbox"/> Does the SCM tool support the automatic notification of success or failure of an action to relevant parties? <input type="checkbox"/> Is it possible to get an overview of all finalised, ongoing and further tasks at the different sites? <input type="checkbox"/> Is it possible to get information about the efforts of project members, i.e. the usability of project members? <input type="checkbox"/> Is it possible for project members to get an overview of further tasks and their estimated efforts and schedule?

Communication	<ul style="list-style-type: none"> <input type="checkbox"/> Has the project included SCM training, including the SCM tool, procedures, and practices? <input type="checkbox"/> Are there specific lists for mailing, on personnel, etc.? <input type="checkbox"/> Are the forms in the SCM system tailorable? <input type="checkbox"/> Are there any additional means of communication, such as email discussion groups, electronic new servers, and the like, under the SCM system.? <input type="checkbox"/> Does the SCM system contain features for an automatic notification of actions? Is the feature tailorable? <input type="checkbox"/> Are the users of the SCM system aware of distribution? <input type="checkbox"/> Does the SCM system have external email integration?
Security	<ul style="list-style-type: none"> <input type="checkbox"/> Have any security rules been defined for the VSC project? <input type="checkbox"/> Are there any specific security restrictions? <input type="checkbox"/> Does the SCM system have access control for SCIs according to sites, roles, and users? Is it possible to define security levels such as no rights, read-only, and right to make changes? <input type="checkbox"/> Is it possible to use compression and encryption when transmitting data? <input type="checkbox"/> Is it possible to restrict the access to the SCM system for specific types of transmission? <input type="checkbox"/> Is it possible to use secure signatures when distributing SCIs?
Management	<ul style="list-style-type: none"> <input type="checkbox"/> What kind of management practices and procedures exist? <input type="checkbox"/> Do the management practices support and follow-up the implementation of SCM procedures, such as switching of tasks, communication, status of the project, etc.

The content and requirements of the non-VSC SCM base processes and SCM plans are well defined in various standards, process assessment and improvement approaches, and other publications. Therefore, they are not dealt with to any larger extent here. When analysing the base practices, the expanded SCM process description can be utilised in the same way as described for teamwork processes above. Furthermore, The SCM-oriented process assessments based on CMM, Bootstrap, and ISO 15504, for example, can easily be utilised for the base processes.

5.4 The Goals for Improved SCM

While the first SCM-Pr²imer phase focuses on analysing the current status of the virtual SCM process, the second phase identifies the actual goals of the SCM improvement program. In addition to the goal definition, the phase includes the tasks of measurement planning and prescriptive process modelling, along with an analysis of the SCM requirements. The steps of the second phase of SCM-Pr²imer are illustrated in Figure 8. Quite like Figure 7, Figure 8 shows similar steps to those found in the standard Pr²imer except for the activities concentrating on the SCM process. As presented below, the activities of the second SCM-Pr²imer phase are based on the GQM approach (Solingen van & Berghout 1999).

SCM-Pr²imer utilises GQM (Goal/Question/Metric), the goal-oriented measurement paradigm (Basili et al. 1994, Solingen van & Berghout 1999). The identification of improvement objectives and measurement goals, and measurement planning are based on the GQM approach. The approach has been described in several publications in the literature, and therefore the method has not been introduced here in detail. A systematic guide for applying GQM is provided by Solingen's and Berghout's book "The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development" (Solingen van & Berghout 1999), for example. For increasing understanding of the goals and SCM requirements, the target SCM processes are defined. Here we enlarge the basic Pr²imer framework by applying the use case approach described in Chapter 3. Generally, a use case is targeted on a specific SCM sub-process. Thus, the expanded SCM process description can be used to help define the content of the use case. On the other hand, the expanded and detailed SCM process description offers a framework for use cases and identified requirements. This ensures the traceability of requirements, as well as an organised structure for results. A systematic approach supports the use, reuse, and maintainability of the SCM process models.

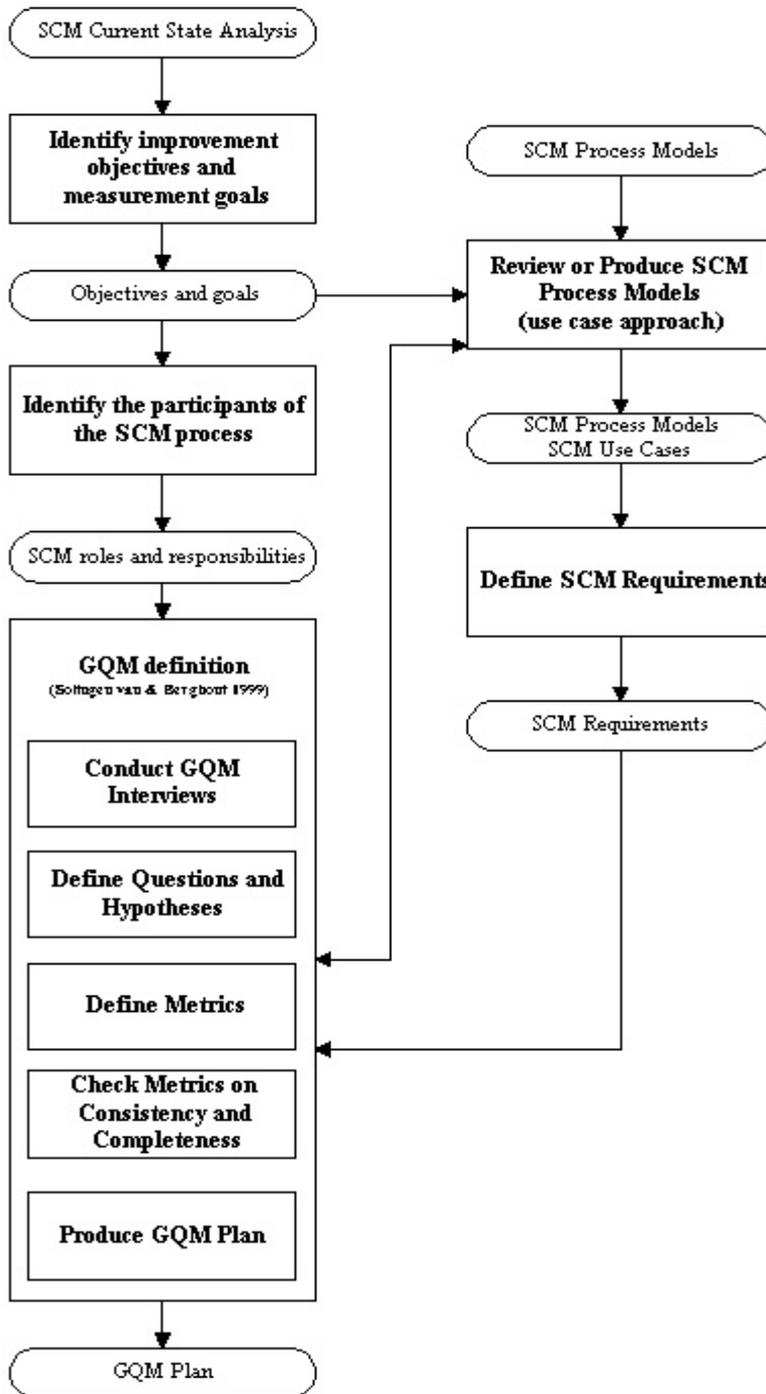


Figure 8. The steps of the second phase of SCM-Primer.

5.5 Plan for SCM Improvement Steps

The third phase of SCM-Pr²imer consists of planning and implementing the improvement and measurement activities (see Paper III). In other words, SCM process improvement and measurement actions are refined into a series of practical tasks, responsibilities and associated schedules. The metrics collection can be particularly laborious within a VSC. For example, in our case study, it has been difficult to acquire defined SCM metrics due to the lack of suitable tools and methods. In addition, the SCM tools examined in our case study do not include all the data needed for analysing the SCM within a virtual environment. LIFESPAN™ and ClearCase®, for example, do not consider the data concerning site and organisation. Due to a number of problems in the metrics definition and collection phase in the VISCOUNT project, we started to investigate and to develop an SCM measurement environment suitable for the VSC environments (see paper VI). In general, the planning and implementation of the metrics collection is one of the most exacting tasks in the SCM process improvement program. The steps of the third phase of SCM-Pr²imer are illustrated in Figure 9. The activities are similar to those of the standard Pr²imer, except for the focus of SCM-Pr²imer being on the SCM process.

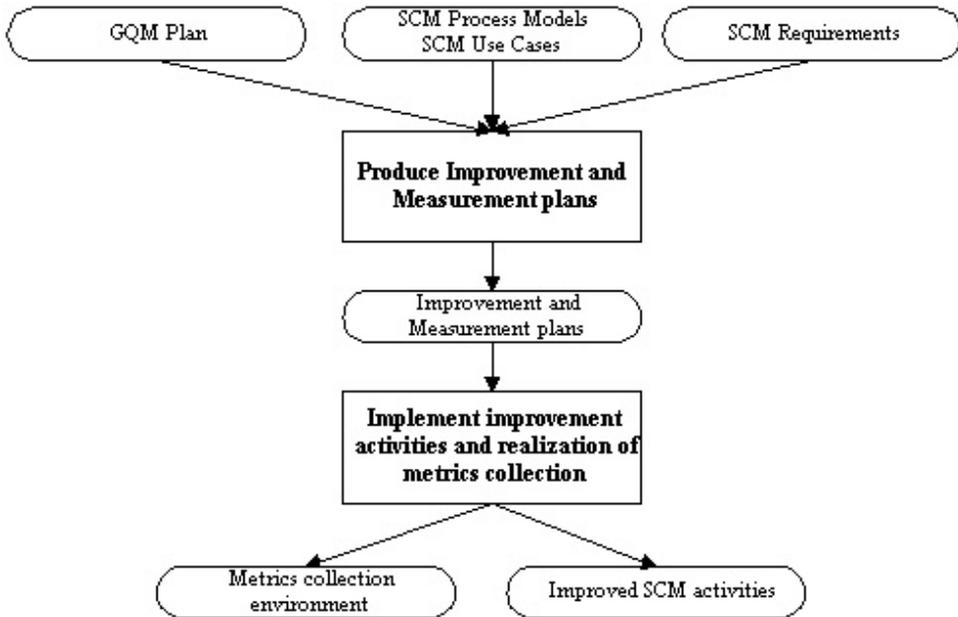


Figure 9. The procedure of the third SCM-Pr²imer phase.

5.6 Pilot Operation and Commissioning

Before taking the revised SCM practices or tools into large-scale use within a VSC, they must be evaluated by testing them in a pilot project or projects. The piloting is carried out according to the improvement plan. Besides piloting, the revised SCM is evaluated in feedback sessions. Here, the collected SCM measurement data is used to make the improvement impacts visible. Generally, the defined SCM metrics can also be used to support the configuration status accounting process, i.e. when analysing and reporting the overall status of the SCM within a VSC. As a basis of the SCM analysis, further actions are planned. In practice, it is decided whether some corrective actions in the improvement program are needed, and if the revised SCM practices or tools can be taken into large-scale use within the complete VSC. Before deploying the procedures for broad use they must be planned carefully and the experiences packaged. Figure 10 illustrates the fourth phase of SCM-Pr²imer. Like the other figures illustrating SCM-Pr²imer phases above, the steps of the Figure 10 are similar to the standard Pr²imer (Karjalainen et al. 1996 and Komi-Sirviö et al. 1998).

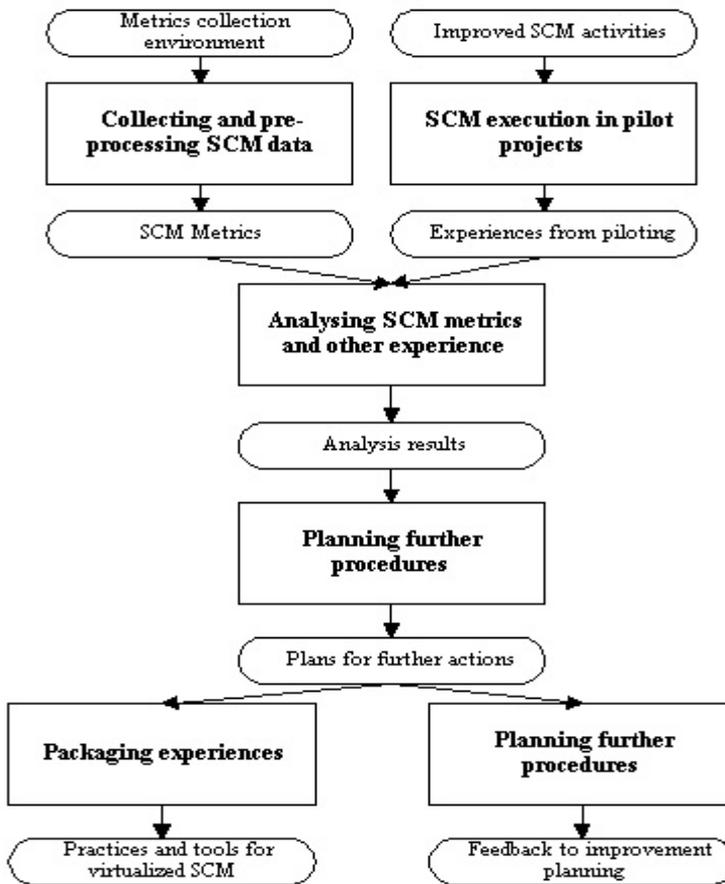


Figure 10. The Steps of the fourth SCM-Pr²imer phase.

5.7 Conclusions

In this chapter we have presented the new SCM-Pr²imer approach to developing SCM in order to meet the VSC-specific requirements. The key requirement for such an approach is comprehensiveness, i.e. the software product, organisation, process, and technical aspects have to be considered in the concept (see *Figure 4*). This case study forms part of a continuing longitudinal study in the quality and process improvement programs of several software organisations (e.g. Komi-Sirviö et al. 1998). The basic structure of SCM-Pr²imer is largely based on the experiences of applying the Pr²imer framework in industrial projects other

than the case organisations of this research. However, during our case study, the content of the approach has been expanded with the aim of supporting SCM within VSCs. The enlargement is seen in the first and second SCM-Pr²imer phases, in which the expanded SCM process description and the use case approach are utilised.

SCM-Pr²imer divides the SCM process improvement into four sequential phases 1) SCM current state analysis, 2) definition of the goals for an improved SCM, 3) plan for improvement steps, and 4) pilot operation and commissioning. The main actions and results, as well as the flow of information from one phase to another, have been described as a short summary in this chapter. The goal of this dissertation is to investigate how SCM can be analysed and developed towards a virtual process. As a result of our research, we are convinced that this goal can be met by analysing SCM in the context of the expanded SCM process description (SCM-Pr²imer phase I). Furthermore, we have described a systematic use case based approach to be used for describing the target state of the SCM process, and defining the VSC-specific requirements for SCM (SCM-Pr²imer phase II, see also Chapter 3). The validation of the SCM analysis results of our application organisations is complete, with highly encouraging results. These results are discussed in more detail in Papers I, II, III, and V. In addition, the use cases have been utilised successfully as a communication tool when defining the target state of the SCM process and also as a support for the SCM tool evaluation.

6. Validation

6.1 Introduction

Chapter 1 of the dissertation describes the background and states the research strategy and the problem to be answered. Chapter 2 introduces the related work done on this subject and summarises the basic concepts related to SCM. Furthermore, the attributes and characteristics of VSCs are discussed in this chapter. In Chapter 3 an approach for defining SCM requirements in the context of our case study is presented. On the basis of the literature survey, the case studies, and the identified SCM requirements, an expanded SCM process description is presented in Chapter 4. In order to establish answers to the research problem, a framework for SCM development towards a virtual process is presented in Chapter 5.

The research results presented in this dissertation are founded on the analysis of the SCM processes and requirements in three industrial organisations, and on a literature survey of related work. In this chapter, the research results are examined and validated in the context of a case study. The research strategy of this dissertation is case study research. The strategy and the research tasks are presented in Chapter 1 Section 1.3. For the validation of the results of a case study research, Yin (1991) presents four tests for judging the quality of results in his book "Case study research: design and methods". The tests, external validity, reliability, internal validity, and construct validity examine the research results from different viewpoints. These tests cover the various aspects extensively when validating the results of a case study research.

In Figure 11 the tests are illustrated by the grey arrow-boxes, and the grey rectangles represent the practical implementations of the tests. These have been adapted from Yin (1991). The rounded squares indicate the most important information sources of this dissertation i.e. the application organisations of the study and related work.

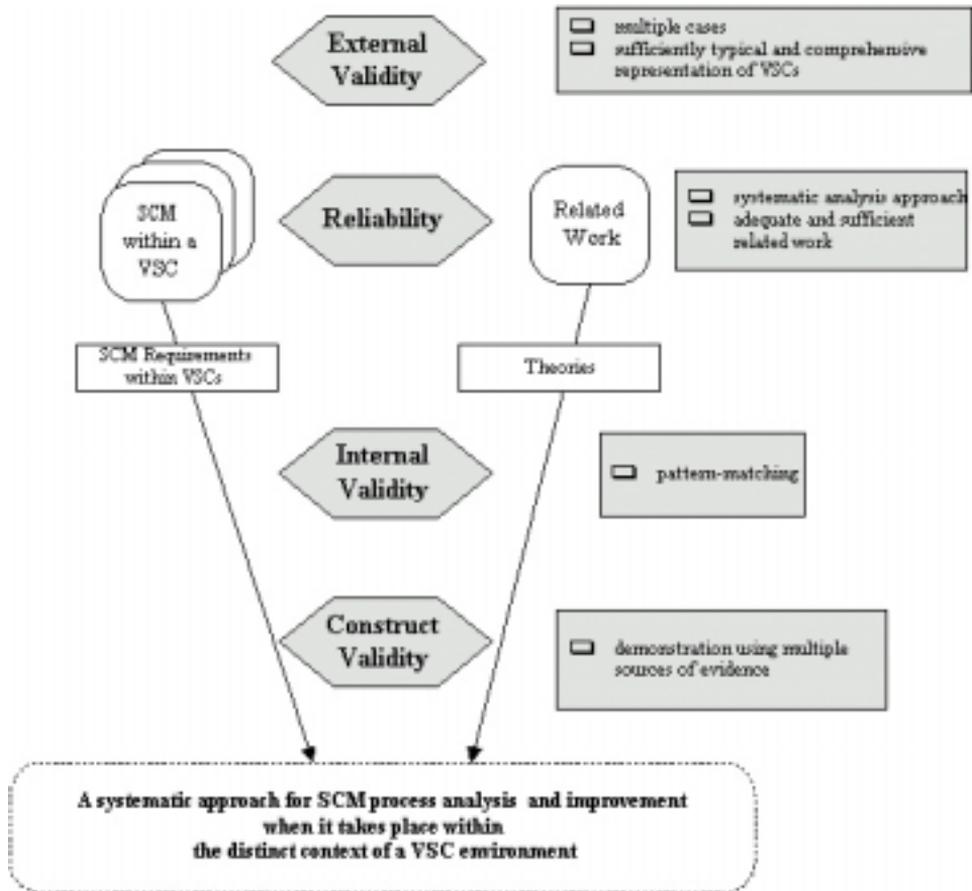


Figure 11. Validation tests of the case study.

The framework, illustrated in Figure 11, is used to demonstrate the validity and contribution of the work of this thesis. However, although the causes of external validity, internal validity, and reliability are significant, the most important goal of this chapter is to demonstrate that the expanded SCM process description and the new SCM-Pr²imer approach can provide adequate support for SCM process improvement within the distinct context of a virtual environment. In other words, the focal point of the validation is the construct validity test, which is discussed in Section 6.3. The validation of the approach and the methods used in this thesis, i.e. the other tests, are discussed briefly in the next section.

6.2 Validation of the Approach and Methods

6.2.1 Internal Validity

The objective of the internal validity test is to show that "certain conditions (in our case: a virtual environment) are shown to lead to other conditions (in our case: new SCM challenges)" (Yin 1991). In other words, it has to be justified that the virtual organisational structure causes special changes to the SCM process, and thus a new definition of the process is needed for analysing and improving it. For internal validity, the pattern-matching analysis is used. The purpose of pattern-matching is to compare the empirical results of the case study with predicted results (Yin 1991). In our study, the VSC characteristics and the identified SCM requirements of three application organisations have been compared against each other, and, additionally, against the characteristics and SCM requirements that have been documented in the related literature. These have been presented in Chapter 2 (Related Work) and in Chapter 3 (SCM Requirements Analysis in a VSC). Thus, it is evident that the SCM process should be expanded to make it possible to analyse and improve it within the VSC environment in the way discussed in more detail in Chapter 4 Section 4.2.

6.2.2 External Validity

Case study research has been criticised for the fact that the results of a specific research may not be representative and that it may be difficult to distil any generic results from the detailed and specific information of a case study (Potts 1993). The purpose of case study research is not, however, to produce any samples. Rather, the goal is to generalise a particular set of results to extend some theory (Yin 1991). Consequently, the expanded description of the SCM process, VSC characteristics, and the SCM challenges form a baseline that will make it possible to identify other cases to which the results of this dissertation are applicable and can be further utilised. In other words, to ascertain the external validity of this dissertation, the domain to which the findings of the study can be generalised must be established. In our study, the external validity of the research results is justified by virtue of the multiple cases included and by the representativeness of the case environment. The case environment and the industrial cases (debis, Neles, and SIA) are presented in Chapter 3 Section 3.2.

One of the strengths of this dissertation is that the case research is based on three different software organisations, rather than just one single case. Thus, the results of the different phases of the research could be collected from diverse environments and analysed against each other. This made it possible to compare the requirements collected from the diverse sources, and to identify the similarities and divergences of the various virtual environments. These are discussed in more detail in Chapter 4 Section 4.2. As a result, it has been possible to clearly demonstrate how the new SCM challenges reflect the identified VSC characteristics, and how the SCM process description should be expanded to better answer the new needs.

Although the virtual organisation itself is a common concept in the literature, it is difficult to give a specific definition for the organisational structure due to the diversity of the research areas involved. In our study, the issue is considered from the viewpoint of SCM. Hence, our conception of the VSCs is composed of not only the most typical characteristics of VSCs, but also of the SCM challenges those characteristics may present to the organisations. The VSC characteristics and SCM challenges, which are presented and used in our study, are based on the literature survey as well as the author's personal experience from the application organisations. The application organisations of this case study differ from each other in terms of company size, net sales, and portfolio. The overview of application organisations and their current SCM challenges related to the virtual structure and resources are presented in Chapter 3 Section 3.2. As a conclusion, it has been demonstrated that debis, Neles, and SIA together form a sufficiently typical and comprehensive representation of virtual organisations in general.

6.2.3 Reliability

The objective of the reliability test is to ascertain that the operations of the study can be repeated with similar results (Yin 1991). This implies that the research steps and operations have to be clearly documented. In our study, we consider reliability from the viewpoints of the related work and the used approach.

The related work presented in Chapter 2 in this dissertation provides an overview to the key references related to the research problem: traditional SCM process definitions and VSC research. As mentioned earlier, a thorough

overview of the evolution of the SCM has been published (Taramaa 1998). Taramaa's thesis includes a conclusion that industrial organisations within a virtual environment need to be investigated further in order to develop and improve SCM towards a virtual process. No such research has been published before this thesis. As mentioned earlier, several traditional SCM process definitions have been published in different publications. Some more widely known definitions are introduced in a short summary in this dissertation. As the traditional SCM process definition has been established in the literature, and the definitions of the different sources are similar in broad outline, they provided a sufficient and adequate basis for future research. Virtual organisations have also been investigated in various publications exploring the subject from diverse viewpoints. In our study, the work related to virtual organisations has been investigated by comparing the VSC attributes and characteristics presented in the several publications with the characteristics of the application organisations of the study. As a result, two VSC attributes, virtual structure and virtual resources, have been identified as well as the related SCM challenges. The results seem to reflect well the current situation within all the three application organisations. Thus, they provide a sufficient basis for our case study.

One of the most important results of the research reported in this dissertation is the conclusion that the challenges present in current SCM process and tools, as well as the improvement requirements, can be determined by modelling the SCM processes. In our study, the role of the use case descriptions have been to support the VSCs in meeting their individual SCM-specific challenges, to help focus on the most critical tasks, and to identify new SCM requirements. The identified SCM requirements have then been used to establish the expanded SCM process definition. This approach is clearly documented, and illustrated with practical examples, in Chapter 3 of this dissertation. The identified SCM requirements can also be used to justify the reliability of the research: although the analysis has been made on the three different industrial organisations, the most important requirements from the application organisations viewpoints have been similar to those presented earlier in Chapter 3 Section 3.4. However, as also stated above, virtuality may appear in various ways in different organisations. Therefore, the applicability of the expanded SCM process definition may cause some concern. However, the purpose of the research has not been to produce any detailed ready-to-use tool for SCM process development. Rather, the main focus has been on identifying a viable framework

to be used in a VSC when analysing the SCM process and improving it towards virtual configuration management. In other words, the expanded SCM process definition, developed in this dissertation, forms an approach that can be tailored to cover the varying viewpoints of different organisations. In practice, this is realised by analysing the current state of SCM at the beginning of the SCM improvement program and defining organisation specific SCM requirements, as described and guided in this dissertation. The approach is also demonstrated in this thesis in Section 6.3.

Another important approach presented in this dissertation in Chapter 5 is the new SCM-Pr²imer for VSCs, which also offers an operational framework ensuring case study repeatability. SCM-Pr²imer is a generic framework, which offers support for organisations in carrying out SCM development. The approach includes a set of well-defined steps and basic actions from current state analysis to piloting and commissioning. However, SCM-Pr²imer does not offer any detailed fixed set of tools for companies. A complete and ready package would, in any case, be inappropriate, because of the multitude of possible improvement goals and SCM process types within different organisations. On the other hand, the general Pr²imer framework has been used in several SPI projects at VTT Electronics and there are a number of publications available on the experiences gained with the framework, such as Karjalainen et al. 1996, and Komi-Sirviö et al. 1998. In other words, the framework itself has been proven to be repeatable, and thus it can be justified that also the SCM-Pr²imer framework is repeatable. Here, SCM-Pr²imer can be seen as an approach to creating a new insight into SCM process development. Used within VSCs together with learning from experience and adapting SCM tools and methods suitable for the organisation in question, SCM-Pr²imer may increase the understanding of the new demands made by SCM and help to bring VSCs to a new level of SCM in the future.

6.3 Construct Validity

The purpose of the construct validity test is to establish "correct operational measures for the concepts being studied" (Yin 1991). According to Yin, to pass the test of construct validity, the specific types of changes that are to be studied should be selected. This section is concerned with the validation of the expanded SCM process definition and that of the SCM-Pr²imer framework, which is

presented in Chapters 4 and 5. The validation is based on an SCM process improvement program of Neles. The purpose is to demonstrate that the expanded view of the SCM process together with the systematic SCM process improvement approach clearly support VSCs in achieving enhanced SCM activities. These enhancements at Neles have been measured and, thus, evidently realised in form of reduced errors and working time in the release manufacturing of an industrial project. The procedure of the SCM process improvement case study at Neles is described in more detail in Papers I, II, III, and V. This section presents the main results and findings of the program. Moreover, this section presents the current status of the Neles case as of January 2000.

6.3.1 Changes to be Studied

The purpose of this section is to describe a set of the criteria that forms the basis of the validation of the expanded SCM process description and the SCM-Pr²imer framework. The validation process, and thus the SCM improvement program at Neles, has been bounded to cover only the issues that are related to the release manufacturing activities. There are two main reasons for this restriction. First, it is not reasonable to try to improve everything at the same time. Rather, the software process improvement should be based on the incremental approach. Thus, although the findings of the SCM process analysis of this case study have addressed several other improvement areas as well, our study has concentrated on the one main issue. Secondly, the release manufacturing related aspects have been viewed as the most important issue within all the interviewed teams at Neles.

The validation criteria used in this dissertation are based on the SCM process improvement goals set by Neles. Furthermore, the goals are grounded on the results of the analysis of the SCM process. The analysis has been performed in three stages. The main tasks and goals of each phase are described in Chapter 3 Section 3.3.3. The summary of the key findings is presented below. The information is based on the confidential interview report made by the author of this thesis during the VISCOUNT project.

In the case project of Neles, new system releases are made for the system test or for customer delivery approximately twice a month. In addition, system releases are regularly needed to support the implementation phase, e.g. for module- and

integration test. The results of the analysis show clearly that the release manufacturing process has been ineffective. The key points of this analysis are:

- ❑ The building of new releases has been too time-consuming. The ideal time would have been no more than one working day for a release. However, the person responsible for release building had estimated at the end of 1997 that one release process would take about three working days altogether.
- ❑ Old or unaccomplished module versions have been found in several releases during or after the version building and CD production for system test. Sometimes rarely erroneous versions have been sent to a customer project before defects have been discovered.
- ❑ Making changes to an interface program has caused errors almost without exception during the next release building.

The above issues as well as the other findings have been discussed in the feedback sessions with the case project members of Neles. The results have indicated that the root causes behind the ineffectiveness of the release manufacturing have not concerned the basic activities of the release manufacturing process, but rather teamwork related issues mainly. The identification of the SCM challenges as a teamwork issue was clearly a result of using the expanded process definition in the SCM analysis. To specify the causes more clearly, the following SCM process improvement goal has been set (the form of the goal is based on the GQM approach):

Analyse the *release manufacturing process* for the purpose of *improving its effectiveness* from the viewpoint of the *project team* in the context of a *VSC environment*.

More specifically, the goal and thus the criteria for the validation has been to shorten the time of release manufacturing by:

1. reducing errors in the process, and
2. improving the communication between the teams of the project.

To define more explicitly how to do this in practice, the process has been analysed in the context of the expanded SCM process description. Thus, the SCM teamwork sub-process, i.e. the SCM infrastructure, facility, security, communication, culture, and switching of resources, have played a focal role during the analysis. As a result, several SCM process improvement actions have been identified. Some key examples of those actions are presented below (with the related SCM sub-processes in brackets):

- ❑ A new checklist for version manufacturing shall be drawn up to ensure a more systematic process (Release Manufacturing).
- ❑ When a new release is to be done, the teams involved shall be informed, so as to ensure that the relevant versions of the SCIs will be checked in to the Clear Case tool (the SCM tool used at Neles) in time. Hence, a communication procedure will be planned and implemented (SCM Communication).
- ❑ The dependency between changed SCIs and release manufacturing control files shall be such that it can be checked automatically. This requires changes to SCM facility, i.e. new scripts shall be designed and implemented, while taking the different SCM tools used across the teams into account (SCM Facility).
- ❑ The information on whether the change belongs to a release or not shall be easily checked. This requires changes to the fault database system. Moreover, it shall be determined how the fault database can be used from remote sites, or an alternative way planned for checking the information (SCM Facility, SCM Communication).

To summarise, the first and the most important lesson learned from the SCM analysis at Neles is that without the expanded SCM process view, the root causes of the SCM problems of their VSC projects would not have been found. This can also be justified by comparing the analysis results of our case study with the results of an external SPICE assessment made at Neles at the end of 1997. The SPICE assessment had evaluated that the SCM process of Neles was at the established level, and indicated that there were not any consequential problems in the process. The Bootstrap assessment, which has been performed at

the beginning of our case study, had demonstrated similar results. However, the real-life facts show a different situation when the expanded view of the SCM process has been used. Thus, there is a distinct gap between the traditional and the expanded approach of the SCM process analysis.

6.3.2 Validation Process

In the validation process, the basic activities of SCM-Pr²imer have been applied. The SCM-Pr²imer approach and the experiences of its use are described in Papers III and V. In this section, the process is discussed in the context of the validation of the expanded SCM process description and the new SCM-Pr²imer framework.

To realise the required enhancements in the SCM process, the improvement activities first have to be planned and implemented. At Neles, the planning has been done in meetings with the case project members. In these meetings, the former action points, i.e. tasks to be carried out, have also been followed up. Moreover, for evaluating and validating the success of the improvement steps, a measurement plan has been prepared. The measurement data has been defined by refining the improvement goal into questions, and the metrics for answering each question as described in Paper III. Hence, SCM metrics for evaluating and validating the success of the enhanced SCM description and the SCM-Pr²imer framework have been defined in the context of a real-life SCM process improvement program. In other words, clear operational measures for the criteria described in Section 6.3.1 have been specified.

The evaluation criteria and the related measures having been defined, the progress of the SCM improvement program has been followed up by analysing the criteria and the measures in the feedback sessions with the case project members. The phases of the evaluation and validation process are described in more detail in Paper V, and also illustrated in Figure 12.

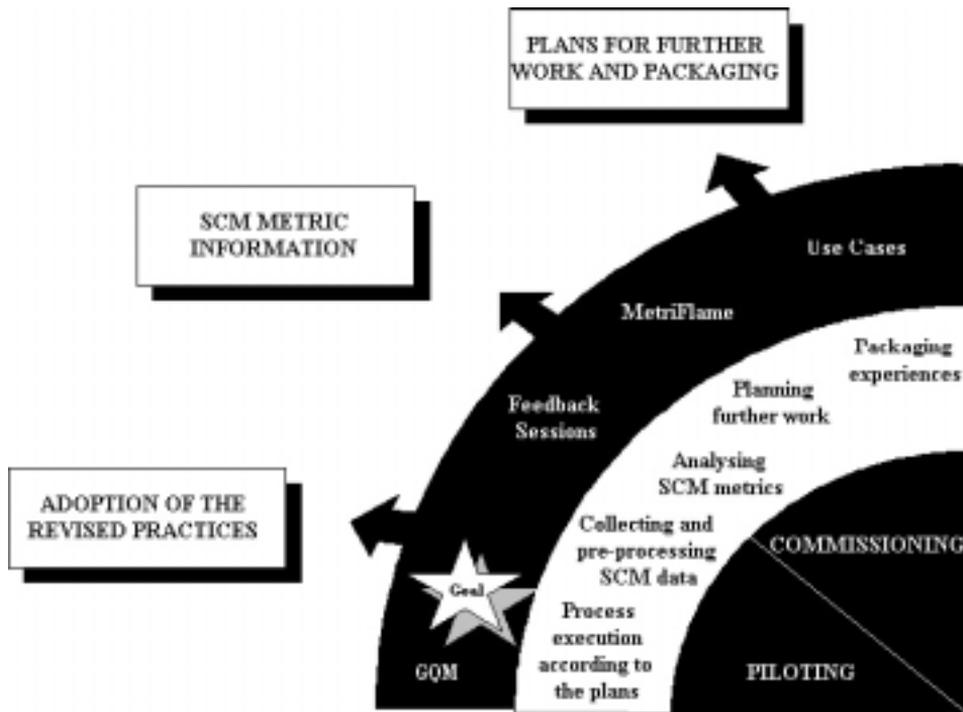


Figure 12. The fourth SCM-Pr²imer phase (Paper V).

6.3.3 Evaluation of the SCM Improvement Program at Neles

In order to evaluate the success of the SCM improvement steps, the SCM metrics were collected and analysed as described in the measurement plan. However, as the SCM data collection within the VSC environment proved to be a more complex task than expected, the amount of the collected data had to be restricted. The complexity has also caused more manual work than expected as most of the SCM data had originally been planned to be collected automatically from SCM tools. However, as the SCM measurement environment has been designed and implemented in parallel, all the required features were not available during the process. This was not, however, critical to demonstrating the success of SCM process improvement at Neles since it was possible to collect the main SCM metrics needed for evaluating the process and the improvement actions. The development of the SCM measurement environment is discussed in more detail in Paper VI. The validation criteria have been analysed in feedback sessions with the case project members of Neles.

Altogether, five feedback sessions have been held between September 1998 and May 1999. Moreover, separate planning and analysis meetings have been arranged occasionally during that time. The analyses have been carried out in the context of the VSC environment.

Table below presents examples of the questions (Q1–Q4) and metrics (M_1.1, M_1.2, etc.). The metrics have been collected regularly and illustrated in the sessions. As a basis of the analysis of the metrics and discussions, it has been discussed if supplementary guidelines, tool support, or some other actions would be needed.

Table 4. Validation questions and metrics.

Q1		What is the number of release manufacturing errors?
	M_1.1	Number of times a release has been rebuilt/delivered.
	M_1.2	Number of added/changed/removed versions of source files in a release in a site.
	M_1.3	Number of errors found on the CD of a release.
Q2		Why do releases have to be rebuilt?
	M_2.1	Number of added/changed/removed source file rows between releases.
	M_2.2	Number of added/changed/removed source file rows between releases by site.
	M_2.3	Number of errors reported by system test group by release.
Q3		How much tool support is required in release manufacturing?
	M_3.1	Number of all support calls related to release building.
	M_3.2	Time spent by tool support personnel on solving version-building problems.
Q4		How much working time does the release manufacturing require?
	M_4.1	Time spent on release manufacturing.
	M_4.2	Time spent on release reporting.

Table presents an example of an SCM metric M_4.1 (time spent release manufacturing) collected during our study. The table presents information on how much time has been used on release manufacturing in our case project. The metric alone does not, however, tell the whole truth; thus a careful analysis is needed. For example, the metric does not include information on how many

releases has been manufactured in one month. Moreover, when a customer delivery is made, it normally requires more time. Thus, it is difficult to make any conclusions on the basis of the table alone. However, as a basis for the discussions with the case project members at Neles and the analysis of the feedback results, it can be justified that the time spent on releasing manufacturing has been reduced during our study. The current evaluation at the end of 1999 is that the release manufacturing per release requires about one day. This is less than 50% of the evaluation at the end of 1997.

Table 5. Time spent on release manufacturing (working days).

Month/ Year	Days
08/98	15
09/98	5
10/98	7
11/98	5
12/98	6
01/99	4
02/99	9
03/99	6
04/99	9
05/99	9

According to the SCM current state analysis, errors have been found on several CD of a release of a system test. The current evaluation at the end of 1999 shows that the number of errors has been reduced. For example, according to the quality meeting minutes 19th of August 1999, the situation has improved and CDs have not included old versions when they have been delivered to the system test. However, a qualitative enquiry made in January 2000 indicates that there are still some problems with the CDs that may cause erroneous situations. Anyway, the same enquiry justifies that the quality of the release manufacturing has been improved and become more uniform. Furthermore, as a result of the actions taken during the study, SCM communication has been enhanced noticeably. For example, more SCM training has been arranged, SCM information practices have been described, and roles and responsibilities have been explicitly defined. Moreover, plans have been made to ensure that all the

teams in all sites have access to the fault database system, which includes information on all faults and change requests. These steps all result from the use of the expanded SCM description and the new SCM-Pr²imer framework.

6.3.4 Conclusions

In the previous sections, the validation criteria, validation process, and the evaluation of the criteria are presented with an example of a real-life case study. The validation is based on the following sources of information: the results of the former SCM process assessment at Neles, the interviews with the case project members of Neles, the collected SCM metrics, the results of the metrics analysis, and the qualitative inquiry made in January 2000.

First, the goal has been to validate the applicability and usability of the expanded SCM process description when analysing the SCM process within a VSC. The results justify the fact that without the expanded view of the SCM process the root causes of the SCM problems would have been difficult to find. Moreover, as indicated by the case project members of Neles, the approach also offers essential support for analysing, defining, and implementing the actions needed to be improved in the process.

Secondly, the goal has been to validate the new SCM-Pr²imer approach itself. The study clearly demonstrates that the approach has provided adequate support to VSCs when improving their SCM process. The measurements clearly indicate that the release manufacturing process at Neles has improved, as the time required for the process has reduced and a number of errors has diminished. This has been achieved through the enhanced activities related to the SCM facility and communication sub-processes. Moreover, the qualitative assessment has shown that the improvements have had a longer-term effect on the working practices at Neles. For example, when starting a new VSC project at Neles, the SCM facility and communication aspects are now routinely considered. This is expected to reduce SCM problems noticeably in the forthcoming VSC projects.

As a conclusion, it can be justified that the expanded SCM process description together with the new SCM-Pr²imer framework provides adequate support for VSCs. Although the approach and especially the question list for SCM analysis need to be modified and developed in the future, even in the present form they

can be used to help researchers to design and to implement more systematic and successful SCM process improvement programs for the VSC environment. Some topics for further research will be discussed in more detail in Chapter 7 of this dissertation.

7. Conclusions

7.1 Research Results

The Virtual Software Corporation (VSC) represents a successful organisational structure addressing the needs of future software developments. Research indicates that Software Configuration Management (SCM) is one of the most important challenges facing VSCs, requiring support and improvement. The aim of this dissertation has been to research, how the SCM process could be analysed, when it takes place within the distinct context of a VSC environment, to provide a basis for improvement. The research problem has been examined in the context of the VISCOUNT project, targeted at developing an SCM tool and creating greater awareness of issues related to the SCM within a virtual environment. The application organisations of the case study are debis Systemhaus, Germany (debis), Neles Automation, Finland (Neles), and Società Italiana Avionica, Italy (SIA).

To provide the basis for the research, the current status of the SCM process within the application organisations has been investigated and a survey of the related literature carried out. As a result, two attributes and four characteristics for VSCs have been defined, corresponding to the key SCM challenges. Our approach recognises the VSC attributes of virtual structure and virtual resources. The virtual structure involves the VSC characteristics of distribution and impermanence. These characteristics provide a powerful concept for enhancing the efficiency of an organisation. However, they also create new challenges related especially to the infrastructure, facility and security issues of SCM. Virtual resources may be human, process-related, or technical assets created and utilised by the organisation. Here, the teams and dynamism characteristics are used for describing the new SCM-related challenges concerning culture, communication, and resource utilisation.

There are a number of approaches that can be used for analysing and improving software processes. Generally, all the approaches consist of an evaluation phase, which determines the current status of the process and, thus, establishes a baseline for the whole improvement cycle and further work. Overall, the analysis of the current state is largely based on the understanding of the content and activities of the process in question. The case examined in this thesis indicates

that the SCM challenges in a VSC are different from those in a traditional single site software organisation, due to the attributes and characteristics of a VSC. Moreover, it is evident that the increasing complexity and the new SCM challenges will result in changes in the SCM process. Traditional SCM process descriptions define the SCM process in broad outline from the viewpoint of a single site environment, without considering the expanded SCM view. Therefore, the traditional procedures of SCM process analysis are not sufficient within a virtual environment. Thus, there has been a clear need to develop an approach to analysing and developing the SCM process when it takes place within the distinct context of a VSC environment. In this dissertation, this issue has been addressed by formulating the general problem into three research questions: Q1) How can the SCM requirements be defined for a VSC?, Q2) How can the SCM process be expanded for a VSC?, and Q3) How can the SCM process be developed to meet the VSC requirements? The main results of this dissertation are discussed below in the context of each of the three research question.

How can the SCM requirements be defined for a VSC? To increase the understanding of the SCM process, specific SCM requirements have been collected and analysed in the context of three VSCs. The analysis has been made by applying the use case technique. The results clearly indicate that the use case approach is a very useful technique for analysing SCM requirements within a VSC environment. Moreover, the use case technique has been successfully used as a part of the SCM process improvement framework in this study. This thesis introduces the approach and presents practical examples of its use in the form that can be utilised within other VSCs as well.

How can the SCM process be expanded for a VSC? As a result of the SCM requirements analysis and literature survey an expanded SCM process definition for VSCs has been defined. Besides the SCM base processes, i.e. configuration identification, configuration status accounting, change control, configuration auditing, and release manufacturing and management, a process concerning SCM Teamwork has been defined. The SCM Teamwork process consists of infrastructure, facility, security, culture, communication, switching, and management sub-processes. The other SCM processes are SCM Planning, with which the base- and SCM Teamwork processes are adjusted to the needs of specific software development projects, and SCM Improvement, by which the

SCM is continuously analysed, measured, and improved within a VSC. As a conclusion, compared with the SCM descriptions presented in current literature, the new expanded SCM process description is more comprehensive, containing elements that have not been identified before. The expanded SCM process description is primarily intended for increasing the understanding of the virtual SCM process and, as such, for supporting the SCM process analysis and providing a basis for SCM process improvement within a VSC.

How can the SCM process be developed to meet the VSC requirements?

Besides the increased understanding of the virtual SCM process, it can be stated that this study has also provided a practical framework to be used in the field of SCM process improvement within the distinct context of a VSC. First, the previous research on new SCM challenges has been highly tool-oriented. In this study, the claim is that the SCM problems within a VSC will not disappear only by acquiring new technology. On the contrary, the SCM process within a virtual environment must be planned, managed, and continuously improved as a complex and dynamically changing system, which forms a critical part of software development. Secondly, SCM is often seen as an internal issue of an organisation. This case study has emphasised the growing number of virtual organisations, where software is developed in a dynamically changing teamwork environment. Under such circumstances, the barriers between organisations are likely to dissolve. Thus, in a virtual environment, more emphasis should be put on planning and maintaining the infrastructure, as well as the facility, security, culture, communication, resource utilisation, and management related issues of SCM, than in a traditional single site environment. The current SCM research is clearly biased towards this view. The new SCM-Pr²imer approach, presented with practical examples in this thesis, utilises the expanded SCM process description. Thus, it forms an advantageous framework for SCM process analysis and development within VSCs.

7.2 Future Research

This thesis has explored SCM within the VISCOUNT project in the context of the three application organisations. However, although the application organisations form a sufficiently typical and comprehensive representation of virtual organisations, it is not an extensive study of this issue. As stated in this

thesis, virtualness and SCM challenges may appear in various ways within different organisations. Thus, although the results of this thesis do increase the understanding of the virtual SCM process, and provide help for determining what needs to be considered when analysing and improving the SCM of a VSC, they do not provide a complete solution to the issue. Thus, subjects for further research might be found in exploring the VSC attributes and characteristics as well as the related SCM challenges and new requirements within other virtual organisations, and comparing the findings to the results of this dissertation. Moreover, it would be advantageous to apply the expanded SCM process description and the new SCM-Pr²imer tool as an VSCs approach within different organisations to test their applicability in large.

As is evident from our case study, the virtual environment considerably increases the need of acquiring more reliable and real-time information on the SCM process. However, as also pointed out in this dissertation, the definition, collection, and analysis of SCM metrics are much more complicated within a virtual than in a single site environment. The SCM process produces a lot of valuable data. A specific item for future research would be to look into how SCM data could be utilised, collected, and analysed within a virtual environment. This study could be carried out as a part of the SCM process improvement program, but also as a consequential element of a more extensive organisational research project. For instance, SCM data definitely has an essential role within knowledge management, which is one of the most important research areas today.

8. Introduction to the Papers

This chapter introduces the original publications included in this dissertation (Papers I–VI). They have all been accepted in scientific volumes of conference proceedings within a two-year period and they are published in this dissertation with the permission of the original publishers.

In Chapter 1 the research questions and strategy of the thesis are introduced. Table 6 presents how the research questions "Q1: How can the SCM requirements be defined for a VSC?", "Q2: How can the SCM process be expanded for a VSC", and "Q3: How can the SCM process be developed to meet the VSC requirements?" are addressed in the papers. Some of the papers are directly focused on a specific question (Table 6) or research task (Table 7). This is referred to in the tables as XX. Furthermore, some of the papers include some aspects related to more than one research question and task (referred to in the tables as X).

Table 6. The original papers of the thesis and the research question.

Paper	Published in	in Proceedings	Q1	Q1	Q1
I	1998	SCM-8	XX	X	X
II	1998	INSPIRE III	X	X	XX
III	1998	UKSMA'98			XX
IV	1998	SPI'98	XX	X	X
V	1999	EUROMICRO 99			XX
VI	1999	EUROMICRO 99			XX

Table 7 gives an overview of how the papers support the research tasks: 1) observe the existing problems and solutions regarding SCM, 2) analyse the SCM requirements in a VSC, 3) analyse and explore a proposal for developing SCM to meet the VSC requirements, and 4) the implementation, evaluation, and validation of the proposal.

Table 7. The original papers of the thesis and the tasks of the research strategy.

Paper	Task 1	Task 2	Task 3	Task 4
I	XX	X	XX	
II		XX	XX	
III		X	XX	X
IV		XX	X	X
V			XX	X
VI			XX	X

8.1 Paper I

Paper I '*Industrial Experiences from SCM Current State Analysis*' discusses the experiences, which have been obtained during the SCM current state analysis in late 1997 within Neles Automation, one of the application organisations of our study (in 1997 the name of the organisation was Valmet Automation, VAT). The paper describes:

- the improvement framework, especially the first phase of SCM-Pr²imer,
- initial idea for the SCM process analysis in the expanded context,
- the experiences from the SCM analysis with the new SCM challenges.

8.2 Paper II

Paper II '*Experiences from Goal Definition for SCM Process Improvement*' describes a systematic approach to the goal definition of software configuration management (SCM) process improvement (the second phase of the SCM-Pr²imer). The approach is based on the goal-oriented measurement paradigm Goal/Question/Metric, GQM (Solingen van & Berghout 1999). Moreover, the experiences gained with the use case approach are presented in the context of Neles Automation, one of the application organisations of our study. The content of the paper can be categorised as follows:

- ❑ the improvement framework, especially the second phase of SCM-Pr²imer,
- ❑ the presentation of the combination of GQM and the use case approach,
- ❑ the presentation of the use case approach in SCM process modelling,
- ❑ an example of SCM goal definition within a VSC.

8.3 Paper III

Paper III *'Practical Experiences from SCM Measurement within a Distributed Environment'* is concerned with a systematic approach to measuring the improvement in the software configuration management (SCM) process. The paper presents the initial results of the SCM measurement program, including the tool support for metric data collection and the analysis of measurement results. The experiences are based on an SCM improvement program within Neles Automation, one of the application organisations of our study. The paper describes:

- ❑ the improvement framework, especially the third phase of SCM-Pr²imer,
- ❑ the experiences from goal oriented SCM measurement,
- ❑ an example of tool support in SCM metrics collection and analysis.

8.4 Paper IV

Paper IV *'Experiences from Requirements Analysis for SCM Process Improvement in Virtual Software Corporation'* provides an introduction to the characteristics of a virtual software corporation (VSC) and the SCM challenges within such an environment. Moreover, the paper describes a systematic approach for SCM process modelling and requirements analysis within a VSC. The main points of the paper are:

- ❑ an introduction to the virtual software corporation (VSC) and SCM challenges,
- ❑ the presentation of applying use cases in SCM process modelling and requirements analysis,
- ❑ the experiences provided by the application organisations of our study.

8.5 Paper V

Paper V '*Experiences from the Pilot Operation and Commissioning Phase of an SCM Process Improvement Program*' discussed the experiences obtained in the analysis and evaluation of an SCM improvement program at Neles Automation one of the application organisations of our study. The paper describes:

- ❑ the improvement framework, especially the fourth phase of SCM-Pr²imer,
- ❑ the experiences from piloting and analysing the improved SCM process,
- ❑ the experiences from SCM metrics collection and the measurement environment.

8.6 Paper VI

Paper VI '*Automating SCM Metric Data Collection and Analysis in Virtual Software Corporations*' presents an initial description of the technical environment for SCM data collection and analysis within virtual software corporations. The paper also describes the experiences of obtaining data from LIFESPANTM to the SCM measurement environment. The main points of the paper are:

- ❑ an introduction to an SCM measurement environment, MetriFlame,
- ❑ *the experiences from the linking of the MetriFlame and LIFESPANTM.*

References

Ahuja, M.K., Carley, K.M. 1998. Network Structure in Virtual Organizations. *Journal of Computer Mediated Communication*. Vol. 3, no. 4. 30 p.

Alstynne, M., van. 1997. The State of Network Organisation: A Survey in Three Frameworks. *Journal of Organizational Computing and Electronic Commerce*. Vol. 7 (2&3), Pp. 83–151.

Asklund, U., Magnusson B., Persson, A. 1999. Experiences: Distributed Development and Software Configuration Management. In: Estublier, J. *System Configuration management, SCM-9*. Toulouse, France: Springer-Verlag. Pp. 17–33. (LNCS 1675.) ISBN 3–540–66484–X.

Auer, A., Taramaa, J. 1996. Experience Report on the Maturity of Configuration Management for Embedded Software. In: Sommerville, I. *Software Configuration Management, SCM-6*. Berlin, Germany: Springer-Verlag. Pp. 185–197. (LNCS 1167.) ISBN 3–540–61964–X.

Basili, V.R., Caldiera, G., Rombach, H.D. 1994. The goal question metric approach. In: Marciniak, J.J (ed.). *Encyclopedia of Software Engineering*. John Wiley & Sons. 1520 p. ISBN 0–471–54004–8.

Birk, A., Järvinen, J., Komi-Sirviö, S., Kuvaja, P., Oivo, M., Pfahl, D. 1998. PROFES - A Product Driven Process Improvement Methodology. *The European Conference on Software Process Improvement (SPI '98)*. 1–4 December. Monte Carlo. 9 p.

Boldyreff, C., Newman, J., Taramaa, J. 1996. Managing Process Improvement in Virtual Software Corporations. *Proceedings of the 5th Workshop on Enabling Technologies: Infrastructure for Collaborative Applications (WET ICE '96)*. Stanford, California: IEEE Computer Society Press. Pp. 142–147.

Brown, N. 1996. Industrial-Strength Management Strategies. *IEEE Software*. July. Pp. 94–103.

Buckley, F. J. 1996. *Implementing Configuration Management, Hardware, Software, and Firmware*. Second Edition. Los Alamitos, California: IEEE Computer Society Press. 380 p. ISBN 0-8186-7186-6.

Buffenbarger, J., Gruell, K. 1999. A Branching/Merging Strategy for Parallel Software Development. In: Estublier, J. *System Configuration Management, SCM-9 Symposium*. Toulouse, France: Springer-Verlag. Pp. 86-99. (LNCS 1675.) ISBN 3-540-66484-X.

CMU/SEI-92-TR-8. 1992. *Configuration Management*, CMU/SEI-92-TR-8. Software Engineering Institute, Carnegie Mellon University. July. Pittsburgh PA.

CMU/SEI-94-HB-04. 1994. *A Systems Engineering Capability Maturity Model, Version 1.0*. Carnegie Mellon University, Software Engineering Institute. December.

Compton, S.B., Conner, G.R. 1994. *Configuration Management for Software*. Van Nostrand Reinhold. 254 p. ISBN 0-442-01746-4.

Conradi, R., Westfechtel, B. 1996. Version Models for Software Configuration Management. *ACM Computing Surveys*. Vol. 30, no. 2, pp. 232-282.

Conradi, R., Westfechtel, B. 1999. SCM: Status and Future Challenges. In: Estublier, J. *System Configuration Management, SCM-9*. Toulouse, France: Springer-Verlag. Pp. 228-231. (LNCS 1657.) ISBN 3-540-66484-X.

Crnkovic, I., Willför, P. 1998. Change Measurements in an SCM Process. In: Magnusson, B. *System Configuration Management, SCM-8*. Brussels, Belgium: Springer-Verlag. Pp. 26-32. (LNCS 1439.) ISBN 3-540-64733-3.

Cugola, G., Fusaro, P., Fugetta, A., Lavazza, L., Manca, S., Pagone, M.R., Ruhe, G., Soro, R. 1997. A Case Study of Evaluating Configuration Management Practices with Goal-Oriented Measurement. *Proceedings of the 4th International Symposium on Software Metrics (Metrics '97)*. Albuquerque, New Mexico, USA. 8 p.

Dart, S. A. 1995. Adopting an Automated Configuration Management Solution. The Eight International Workshop on Computer-Aided Software Engineering, CASE'95. July. Toronto, Ontario. 15 p.

Davidow, W.D., Malone, M.S. 1992. The Virtual Corporation, Structuring and Revitalizing the Corporation for the 21st Century. New York, USA: Harper Business. 294 p. ISBN 0-88730-657-8.

Eman, K., Drouin, J.-N., Melo, W. 1998. SPICE, The Theory and Practice of Software Process Improvement and Capability Determination. Los Alamitos, California: IEEE Computer Society. 486 p. ISBN 0-8186-7798-8.

Estublier, J. 1999. Distributed Objects for Concurrent Engineering. In: Estublier, J. System Configuration Management, SCM-9. Toulouse, France: Springer-Verlag. Pp. 172-185. (LNCS 1675.) ISBN 3-540-66484-X.

Estublier, J., Casallas, R. 1994. The Adele Configuration Manager. Configuration Management, Number 2 in Trends in Software. London, UK: Wiley. Pp. 99-134.

Fitzgerald, B., O'Kane, T. 1999. A Longitudinal Study of Software Process Improvement. IEEE Software. May/June. Pp. 37-45.

Ford, D., Gadde, L.-E., Håkansson, H., Lundgren, A., Snehota, I., Turnbull, P., Wilson, D. 1998. Managing business relationships. Chichester: John Wiley & Sons. 292 p. ISBN 0-471-97075-1.

Frühauf, K., Zeller, A. 1999. Software Configuration Management: State of Art, State of Practice. In: Estublier, J. System Configuration Management, SCM-9. Toulouse, France: Springer-Verlag. Pp. 217-227. (LNCS 1675) ISBN 3-540-66484-X.

Glass, R.L. 1994. The Software Research Crisis. IEEE Software. November, 1994. Pp. 42-47.

Glass, R.L. 1995. A Structure-Based Critique of Contemporary Computing Research. Journal of Systems Software. Vol. 28. Pp. 3-7.

Grady, R.B. 1997. Successful Software Process Improvement. Prentice-Hall. 314 p. ISBN 0-13-626623-1.

Hoek, A. van der., Heimbigner, D., Wolf, A. L. 1996. A Generic, Peer-to-Peer Repository for Distributed Configuration Management. Proceedings of the 18th International Conference on Software Engineering. Berlin, Germany. Pp. 308-317.

Hoek, A. van der, Heimbigner, D., Wolf, A. L. 1998. System Modeling Resurrected. System Configuration Management, ECOOP'98, SCM-8 Symposium. Brussels, Belgium: Springer-Verlag. Pp. 140-145. (LNCS 1439) ISBN 3-540-64733-3.

Håkansson, H., Snehota, I. 1995. Developing relationships in business networks. Routledge, London. 418 p. ISBN 0415115701.

IEEE-1042. 1987. IEEE Guide to Software Configuration Management Plans (ANSI). Piscataway, New Jersey. 92 p.

IEEE-610.12. 1990. IEEE Standard Glossary of Software Engineering Terminology (ANSI). The Institute of Electrical and Electronics Engineers. Piscataway, New Jersey. 83 p.

IEEE-828. 1990. IEEE Standard for Software Configuration Management Plans (ANSI). The Institute of Electrical and Electronics Engineers. Piscataway, New Jersey. 16 p.

ISO/IEC 15504-2. 1998. Information Technology - Software Process Assessment - Part 2: A reference model for processes and process capability. ISO/IEC TR 15504-2.

ISO/IEC-12207. 1995. Information technology - Software Life Cycle Processes. ISO/IEC Copyright Office. Geneva, Switzerland. 57 p.

Jacobson, I. 1995. Use Cases in Large Systems. ROAD. Vol. 1, No. 6, pp. 9-12.

Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G. 1992. Object-Oriented Software Engineering: A Use Case Driven Approach. ACM Press. 350 p. ISBN 0-201-54435-0.

Jones, J., Hewitt, P., Lee, R., Smith, L, Sorensen, R. 1999. Software Configuration Management Technologies and Applications. Software Technology Support Center STSC. STSC Technology Report. <http://www.stsc.hill.af.mil>. 46 p.

Karjalainen, J., Mäkäräinen, M., Komi-Sirviö, S., Seppänen, V. 1996. Practical process improvement for embedded real-time software. Quality Engineering. Vol. 8, No. 4, pp. 565-573.

Kilpi, T. 1998. Applying Product Management Approach to Software Development: An SME Perspective. Oulu, Finland: Acta Universitatis Oulu. 85 p. + app. (A 304.) ISBN 951-42-4792-2.

Kinnula, A. 1999. Software Process Engineering in a Multi-Site Environment. Oulu, Finland: Oulu University Press. 119 p. ISBN 951-42-5302-7.

Komi-Sirviö, S., Oivo, M., Seppänen, V. 1998. Experiences from practical software process improvement. In: Proceedings of the European Software Process Improvement Conference (EuroSPI'98). Gothenburg, Sweden. Pp. 5.31-5.45.

Kuvaja, P., Similä, J., Krzanik, L., Bicego, A., Saukkonen, S., Koch, G. 1994. Software Process Assessment & Improvement: The BOOTSTRAP Approach. Great Britain: Blackwell Publishers. 149 p. ISBN 0-631-19663-3.

Larsen, J-O., Roald, H.M. 1998. Introducing Clear Case as a Process Improvement Experiment. In: Magnusson, B. System Configuration Management, SCM-8. Brussels, Belgium: Springer-Verlag. Pp. 1-12. (LNCS 1439). ISBN 3-540-64733-3.

Lee, W.J., Cha, S.D., Kwon, Y.R. 1998. Integration and Analysis of Use Cases Using Modular Petri Nets in Requirements Engineering. IEEE Transactions on Software Engineering. Vol. 24, No. 12, pp. 1115-1126.

Lipnack, J., Stamps, J. 1997. *Virtual Teams: Researching across space, time, and organizations with technology*. New York: John Wiley and Sons, Inc. 262 p. ISBN 0-471-16553-0.

McGowan, S. 1997. *VISCOUNT Project Programme, EU/Esprit project #25754*. Version 4.1, 21st of July. 101 p.

Mowshowitz, A. 1997. *Virtual Organization*. *Communications of the ACM*. Vol. 40, No. 9, pp. 30-37.

Noll, J., Scacchi, W. 1997. *Supporting Distributed Configuration Management in Virtual Enterprises*. *Software Configuration Management SCM-7*. Boston, MA, USA: Springer-Verlag. Pp. 142-160. (LNCS 1235) ISBN 3-540-63014-7.

Nonaka, I., Takeuchi, H. 1995. *The knowledge-creating company: How japanese companies create the dynamics of innovation*. New York: Oxford University Press. 284 p. ISBN 0-19-509269-4.

Parnas, D.L. 1998. *Successful Software Engineering Research*. *Software Engineering Notes*. Vol. 23, No. 3, pp. 64-68.

Paulk, M.C., Weber, C.V., Curtis, B., Chrissis, M.B. 1995. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley Publishing Company, Inc. 441 p. ISBN 0-201-54664-7.

Potts, C. 1993. *Software Engineering Research Revisited*. *IEEE Software*. September 1993. Pp. 19-28.

Pressman, R.S. 1993. *A Manager's Guide to Software Engineering*. McGraw-Hill, Inc. 528 p. ISBN 0-07-050820-8.

Seppänen, V., Kähkönen, A.-M., Oivo, M., Perunka, H., Isomursu, P., Pulli, P. 1996. *Strategic Needs and Future Trends of Embedded Software*. *Technology review* 48/96. Sipoo, Finland: Paino-Center Oy. 103 p. ISBN 951-53-0744-9.

Solingen van, R., Berghout, E. 1999. The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development. London, UK. McGraw-Hill Publishing Company. 199 p. ISBN 007-709553-7.

Stähle, P., Grönroos, M. 1999. Knowledge Management - Tietopääoma yrityksen kilpailutekijänä (Knowledge Management - a Competitive Advantage for Business). Porvoo: WSOY. 218 p. (In Finnish). ISBN 951-0-23591-1.

Sutcliffe, A.G., Maiden N., Minocha, S., Manuel, D. 1998. Supporting Scenario-Based Requirements Engineering. IEEE Transactions on Software Engineering. Vol. 24, No. 12, pp. 1072-1088.

Taramaa, J. 1998. Practical development of software configuration management for embedded systems. Espoo, Finland. Technical Research Centre of Finland. 147 p. + app. 110 p. (VTT Publications 366.) ISBN 951-38-5344-6.

Tichy, W. 1988. Tools for Software Configuration Management. Proceedings of the International Workshop on Version and Configuration Control. Grassau. Pp. 1-20.

Van De Vanter, M. L., Murer, T. 1999. Global Names: Support for Managing Software in a World of Virtual Organizations. In: Estublier, J. System Configuration Management, SCM-9 Symposium. Toulouse, France: Springer-Verlag. Pp. 157-171. (LNCS 1675.) ISBN 3-540-66484-X.

Wein, M., Mackay, S.A., Stewart, D.A., Gentleman, W.M. 1995. Evolution is Essential for Software Tool Development. Proceedings of the Seventh International Workshop on Computer-Aided Software Engineering. Toronto, Ontario, Canada. Pp. 196-205.

Venkatraman, N., Henderson, J.C. 1998. Real Strategies for Virtual Organizing. Sloan Management Review. Vol. 40, No. 1, pp. 33-48.

Yin, R.K. 1991. Case study research: design and methods. USA: Sage Publications Inc. 165 p. ISBN 0-8039-3470-X.

***Appendices of this publication are not included in the PDF version.
Please order the printed version to get the complete publication
(<http://otatrip.hut.fi/vtt/jure/index.html>)***

Published by



Vuorimiehentie 5, P.O.Box 2000, FIN-02044 VTT, Finland
Phone internat. +358 9 4561
Fax +358 9 456 4374

Series title, number and
report code of publication

VTT Publications 409
VTT-PUBS-409

Author(s) Rahikkala, Tua			
Title Towards virtual software configuration management A case study			
Abstract <p>The organisational performance of software companies has become critically important. While customer requirements are changing and varying ever more frequently, and an effective management of the software process is becoming more and more essential, the appropriateness of current software development models has become questionable. One of the current global trends in software development is transorganisational collaborative work in distributed, dynamic teamwork environments, called <i>Virtual Software Corporations (VSC)</i>. The virtual environment presents particular challenges to <i>Software Configuration Management (SCM)</i>, while the rate of change concerning the size, complexity and duration of software projects is increasing constantly. While there is a clear need for VSCs to analyse and improve their SCM processes, they may approach the subject from rather different angles. Generally, when analysing the current status of the software process, it is vital to understand what the context of the process is and what activities it includes. On the other hand, our study makes it evident that VSCs present new kinds of challenges to SCM that can not be solved by means of traditional SCM procedures and techniques only. Furthermore, many of the VSC variants are likely to realise that the increasing complexity and number of their software processes also affect the SCM process directly. Hence, a sound understanding of the specific context is required for analysing and improving the SCM in a VSC. Gaining extensive knowledge of the SCM-related requirements for industrial VSCs can thus be considered a prerequisite for enhancing the process description. This dissertation presents an approach for defining the SCM requirements for a VSC. On the basis of a requirements analysis, an expanded SCM process description is introduced. As a result, this dissertation introduces an enhanced approach to SCM process analysis and improvement for the distinct context of a VSC environment.</p>			
Keywords Software Configuration Management (SCM), Virtual Software Corporation (VSC), distributed teamwork environment, dynamic teamwork environment, Software Process Improvement (SPI), SCM process analysis			
Activity unit VTT Electronics, Embedded Software, Kaitoväylä 1, P.O.Box 1100, FIN-90571 OULU, Finland			
ISBN 951-38-5567-8 (soft back ed.) 951-38-5568-6 (URL: http://www.inf.vtt.fi/pdf/)		Project number	
Date May 2000	Language English	Pages 110 p. + app. 57 p.	Price D
Name of project VISCOUNT		Commissioned by EU, National Technology Agency (Tekes), VTT Electronics	
Series title and ISSN VTT Publications 1235-0621 (soft back ed.) 1455-0849 (URL: http://www.inf.vtt.fi/pdf/)		Sold by VTT Information Service P.O.Box 2000, FIN-02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374	